



Universitat
Autònoma
de Barcelona



ESTUDI DE COMPORTAMENTS SOCIALS AMB EIXAMS DE ROBOTS RECONFIGURABLES

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Alberto Jiménez Riera
i dirigit per
Lluís Ribas Xirgo
Bellaterra, 16 de setembre de 2009

El sotasignat, Lluís Ribas Xirgo

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Alberto Jiménez Riera

I per tal que consti firma la present.

Signat:

Bellaterra,de.....de 200.....

Taula de continguts

Capítol 1 Introducció i Estudi previ.....	3
1.1 Objectius del projecte.....	3
1.2 Breu introducció a l'estat del art del tema proposat.....	4
1.3 Estudi de viabilitat del projecte.....	5
1.4 Planificació temporal del treball.....	6
Capítol 2 Aplicació.....	8
2.1 Tria d'un escenari realista per a la simulació.....	8
2.2 Tria d'un robot adequat.....	12
2.2.1 Requeriments: formes d'eliminar insectes.....	12
2.2.2 Tria de components.....	14
2.3 Consum energètic.....	16
2.3.1 Temps necessari per a cada tasca.....	17
2.3.2 Consum energètic necessari per a cada tasca.....	18
Capítol 3 Simulació i Modelat de l'Entorn.....	20
3.1 Simulador original: R&B Sim.....	20
3.2 Modificacions del simulador R&B Sim.....	22
3.2.1 Modelat de l'entorn.....	22
3.2.2 Modelat del comportament dels robots: estratègies.....	24
Capítol 4 Codificació de les adaptacions.....	29
4.1 Adaptacions al modelat inicial.....	30
4.1.1 Distribució inicial de les plantes.....	29
4.1.2 Distribució inicial dels robots.....	30
4.1.3 Distribució inicial dels insectes.....	32
4.2 Adaptacions a les estratègies de cerca.....	33
4.3 Adaptacions a les polítiques de reconfiguració.....	36
4.4 Diagrama de la màquina d'estats adaptada.....	37
Capítol 5 Simulacions.....	39
5.1 Estudi de l'efectivitat: nombre de punts treballats.....	41
5.2 Consum: anàlisi de la despesa energètica mitjana acumulada dels robots.....	44
5.3 Nombre de reconfiguracions: causa de les diferències de consum energètic.....	49
Capítol 6 Conclusions.....	51
6.1 Assoliment dels objectius.....	51
6.1.1 Treball realitzat.....	51
6.1.2 Seguiment de la planificació temporal.....	52
6.2 Anàlisi dels resultats i consideracions importants.....	53
6.3 Valoració final.....	54
Referències.....	57

Capítol 1

Introducció i Estudi previ

1.1 Objectius del projecte

Aquest treball és un estudi dels comportaments socials d'un conjunt de robots que treballen alhora per solucionar un problema. El fet que es desitgi estudiar el comportament social, implica que serà necessari disposar d'un eixam relativament gran de robots. Una de les possibles aplicacions que pot tenir aquest estudi és la inclusió de noves regles de comportament social que resultin útils a l'hora de solucionar un problema de cobertura d'un terreny, com pot ser el problema tipus que s'anomena el joc del "Red and Blue"[14]. En aquest joc, s'hi poden trobar entorns que contenen punts de color vermell i punts de color blau. Es disposa d'uns robots capaços de realitzar operacions sobre els punts d'un color determinat i d'aquesta manera neutralitzar-los (o canviar-los a un color negre). Cadascun d'aquests robots és capaç de treballar sobre qualsevol dels dos punts, però el fet de passar de treballar sobre punts vermells a treballar sobre punts blaus implica un cost addicional. El joc acaba quan els robots han treballat sobre tots els punts i, per tant, no queden ni punts vermells ni punts blaus.

Es podria pensar que hi ha moltes solucions vàlides per acabar el joc i neutralitzar tots els punts però, tot i que pugui ser cert que moltes d'elles podrien conduir al mateix nombre de punts neutralitzats, també podrien tenir un cost diferent. És precisament aquest factor el que determina la bondat d'aplicar unes o altres regles de comportament i això en condicionarà el resultat.

Per tal de dur a terme aquest estudi cal acotar aquest joc a un entorn menys abstracte. Hom podria pensar en una aplicació d'aquest joc per al control de plagues[7] en entorns agraris. Així, els punts que ja hem esmentat serien dos tipus d'insecte diferents i els robots recorrerien un camp tot eliminant mecànicament els insectes d'una mena o altra. Això resulta així pel fet que l'eliminació d'un tipus d'insecte i de l'altre normalment seran procediments diferents i implicaran una utilització diferent de sensors i actuadors del robot.

El fet de reconfigurar-se per passar de matar un tipus d'insecte a un altre, implica un cost en temps i en energia. Per tant, caldrà definir estratègies tant de cerca dels insectes com de reconfiguració. Utilitzant diferents estratègies els robots podran assolir l'objectiu amb un cost, que

podrà ser comparat per avaluar l'eficàcia de cadascun d'aquests comportaments. Aquesta aplicació en concret condicionarà la tria d'un robot capaç de desplaçar-se en un entorn d'aquest tipus i resistent a les inclemències que podria sofrir en un entorn agrari a l'aire lliure.

Disposar d'un eixam força ampli que permeti extreure conclusions de comportaments socials amb robots d'aquesta mena, resultaria molt costós, és per això que aprofitarem una de les eines que ens permet la informàtica: un simulador. Com qualsevol simulador, treballarà amb un model simplificat d'aquests robots i de l'entorn on es mouran i per tal de fer-ho serà necessari disposar d'aquest model.

En aquest punt esdevé necessari poder determinar els paràmetres del robot triat, que determinaran els costos associats al moviment del robot, a l'eliminació d'insectes i a la reconfiguració de sensors i efectors. Una vegada obtinguts aquests costos, caldrà modelar l'entorn al que estarà sotmès el robot. Això vol dir que es simularà el comportament de l'eixam en un escenari que imitarà un entorn amb plantes i dos tipus d'insecte diferents. En executar la simulació es poden obtenir els resultats amb els costos que tenen associat i serà precisament aquest resultat el que permetrà fer les comparacions necessàries entre comportaments diferents.

En resum, aquest treball busca obtenir un model de comportament social el més òptim possible per a l'eliminació de plagues en entorns agraris mitjançant robots. El seguit de passos per aconseguir-ho serà: modelar el problema, crear els escenaris on estudiar aquest comportament, crear un model de variabilitat del comportament i obtenir els patrons de les dinàmiques dels comportaments socials que s'observin.

1.2 Breu introducció a l'estat del art del tema proposat

Els estudis de comportaments socials darrerament es duen a terme en molts camps. S'originaren a les ciències socials però els investigadors s'han adonat poc a poc que resulten també d'especial interès en l'àmbit tecnològic. Aquest fet que podria resultar sorprenent es justifica ja que a la robòtica es poden trobar societats de robots que actuen conjuntament per assolir un objectiu i en especial resulta d'una importància cabdal el comportament social en els eixams de microrobots.

Es poden entendre els microrobots de diverses maneres. Per a determinats sectors un microrobot és un robot capaç de realitzar

operacions amb una precisió de micròmetres mentre que d'altres sostenen que un microrobot és un robot d'una mida micromètrica. Aquest estudi prendrà el segon significat en referir-se a microrobots. En aquest tipus de robots la limitada capacitat de càlcul obliga a l'adopció d'estratègies socials per tal de solucionar un problema. La idea de fons consisteix en adoptar un seguit de regles que permetin, de la mateixa manera que succeeix en la nostra societat, organitzar-se de tal forma que entre tots assoleixin un objectiu prèviament fixat.

Els microrobots podrien resultar molt interessants en el camp de la medicina. Actualment hi ha molts tractaments de determinades patologies que impliquen un tractament massiu. Un exemple clar n'és el tractament amb antibiòtics, que arrasa literalment els bacteris de l'intestí i fa minvar les defenses de les mucoses, però també es veu clarament amb els efectes perjudicials de la quimioteràpia en tractaments de càncer. Si es disposés d'uns robots de mida prou reduïda com per poder desplaçar-se per fluids i teixits del cos humà, es podria fer un tractament més localitzat. En el cas del càncer, es podria tractar únicament aquella cèl·lula que sofrís la mutació, evitant d'aquesta manera multitud d'efectes secundaris del tractament. No obstant, aquest món no és tan proper com es podria pensar. Els microrobots creats fins ara no tenen encara una utilitat massa clara en biomedicina, ja que encara són de dimensions massa considerables per poder-se fer servir a l'interior del cos humà.

Actualment també està de moda parlar de nanorobots, que encara és quelcom més subjectiu. Hi ha entitats que es dediquen a fer investigacions sobre possibles aplicacions, sobretot mèdiques, de robots de dimensions d'uns quants nanòmetres[6]. Seria ideal si un robot d'aquestes dimensions fos plausible, però malauradament les lleis de la física quàntica farien que aquests robots no es comportessin com a sòlids i seria del tot impossible predir el comportament mitjançant simulacions, fent-se necessàries multitud d'experimentacions diferents.

Tot i les aparents dificultats de la microrobòtica, els resultats d'aquest estudi poden resultar útils en un futur per a l'aplicació d'estratègies socials en microrobots més avançats que de ben segur estaran disponibles en un temps relativament breu.

1.3 Estudi de viabilitat del projecte

La viabilitat del projecte és un aspecte a tenir en especial consideració tractant-se d'un estudi d'aquest tipus. En primer lloc, ja s'ha vist que la possible utilització d'aquesta mena de comportaments encara queden ben lluny de ser plenament útils per a la majoria d'aplicacions reals actuals. No obstant, cal destacar que els objectius que s'han fixat no són estàtics i es poden reduir o ampliar en funció dels resultats que es vagin

obtenint. En aquest sentit, entenent la viabilitat com a la possibilitat d'assolir els resultats previstos a l'estudi, el projecte resulta plenament viable. Si es considera la viabilitat com a possibilitat de dur a la pràctica el resultat de l'estudi en una aplicació real, resulta més aviat dubtosa. D'una banda per que les simplificacions de l'entorn i dels robots podrien fer minvar la certesa dels resultats i de l'altra perquè hi ha altres estratègies més adequades que aquest tipus de robots ens permetrien utilitzar (sense les limitacions que imposaria la utilització de microrobots).

Quant a requeriments materials i econòmics, no resulten un problema en aquest projecte. Per tal de minimitzar costos, es disposa d'un simulador realitzat en Visual C++ que simula el joc del "Red and Blue" que, convenientment modificat permet simular l'eliminació de plagues amb robots. Això implica que cal disposar d'un ordinador capaç d'executar el Microsoft Visual C++, que també implica disposar del sistema operatiu Microsoft Windows. Com que es tracta pràcticament d'un estàndard en els ordinadors això no resulta un problema.

Pel que fa a requeriments de personal, serà necessari el treball d'un enginyer informàtic durant aproximadament 450 hores, que seran les necessàries per dur a terme i documentar l'estudi.

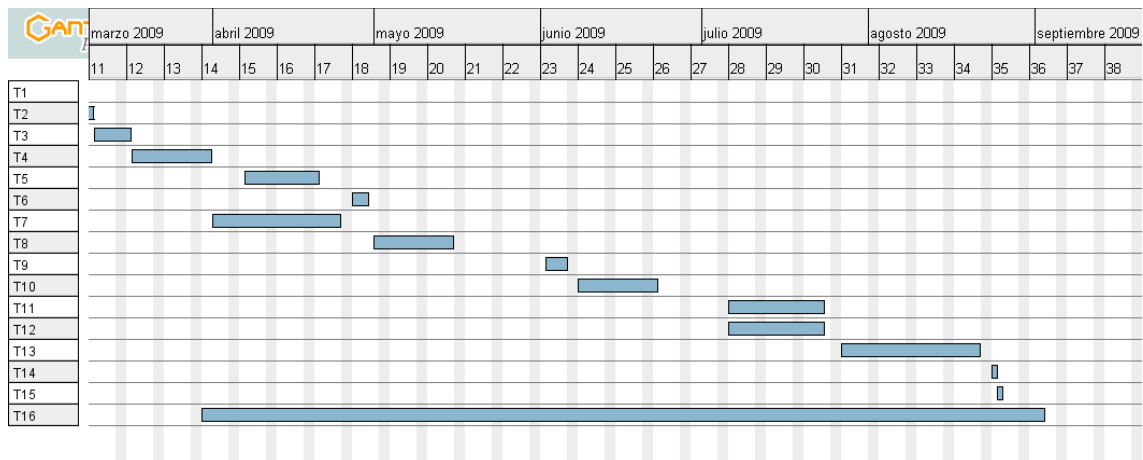
1.4 Planificació temporal del treball

Les tasques a desenvolupar per tal de dur a terme aquest estudi són les següents:

1. Cercar i analitzar informació sobre els comportaments socials en robots (10h)
2. Cercar possibles aplicacions amb microrobots reconfigurables en les que sigui possible aplicar comportaments socials (20h)
3. Determinar una aplicació amb robots per tal de realitzar l'estudi (20h)
4. Buscar informació sobre cultius i l'afectació de les plagues (25h)
5. Elaboració d'un informe previ (20h)
6. Estudiar la distribució de plagues al camp (10h)
7. Buscar i comparar robots adients per desenvolupar la tasca triada (35h)
8. Extreure els paràmetres funcionals i no funcionals dels robots triats (25h)

9. Estudiar el simulador d'eixams de microrobots (20h)
10. Fer petites modificacions al simulador (50h)
11. Buscar estratègies de cerca aplicables als robots (35h)
12. Buscar estratègies de reconfiguració aplicables als robots (35h)
13. Inserir al simulador les diferents estratègies de cerca i de reconfiguració (50h)
14. Comprovar la bondat de cadascuna de les configuracions anteriors (5h)
15. Escollir el comportament que solucioni el problema de forma més eficient (5h)
16. Elaboració de la memòria del projecte (85h)

La distribució inicial en el temps d'aquestes tasques es pot veure al diagrama de Gantt següent:



Capítol 2

Aplicació

Hi ha moltes formes de dur a terme un estudi de comportaments socials, depenent de quin sigui el camp d'aplicació. A les ciències socials s'acostuma a observar el comportament de la població, i a les ciències naturals el d'un conjunt d'individus d'una o més espècies. En el camp que ens ocupa, que és el de la robòtica, seria possible construir uns robots, programar-ne uns comportaments i observar el resultat en ubicar-los en un entorn concret; però el cost d'aquesta solució seria molt elevat i seria molt difícil provar diversos comportaments diferents en un període de temps prou breu. L'altra alternativa, la més factible, consisteix en simular el comportament dels robots mitjançant una aplicació informàtica. D'aquesta manera, comprovar el resultat de diferents polítiques aplicades als robots resulta molt més ràpid i fàcil.

Estudiar els comportaments socials sense un entorn ni un objectiu fixats no conduiria a cap resultat útil. Seria possible anar modificant comportaments, afegint i traient robots del simulador i observant alguna mena de patró de comportament, però no serviria per solucionar cap problema real i tampoc resultaria útil per determinar l'efectivitat d'un o altre comportament. Per tal de poder comparar els comportaments és necessari cenyir-se a un determinat problema. Precisament aquest aspecte es tracta al capítol actual. Es determinarà quin problema resoldran els robots per tal de poder dissenyar, finalment, el simulador.

2.1 Tria d'un escenari realista per a la simulació

Per realitzar l'estudi és convenient trobar un escenari sobre el que poder dur a terme les simulacions. Centrant-nos en robots agricultors es poden tractar diversos temes però ocupem-nos de l'eliminació de plagues en un cultiu determinat.

El tipus de cultiu a triar pot ser molt variat, des de plantes lleguminoses fins a fruites, passant pels cereals. Per exemple, les tomaqueres poden ser atacades per erugues i aranyes, entre d'altres insectes [1][11]. No obstant això, que s'ajustaria a la classe de problemes a resoldre amb robots multi-funcionals, cal tenir present que haurien de poder treballar en un rang d'alçades de fins a tres metres, atesa l'alçària a la que solen arribar les tomaqueres. Per tal de no restringir el tipus de robot a utilitzar és preferible centrar-se en plantes més baixes com les cigroneres, que

també poden ser atacades per plagues diverses i, a diferència de les tomaqueres, són més baixes.

La cigronera, al contrari que la tomaquera, és una planta que arriba a fer uns 50-60 centímetres[10][8], el que deixa força llibertat pel que fa a la tria del robot que hi treballarà. Els seus fruits, els cigrons es formen dins d'una beina en la qual acostumem a trobar generalment dos cigrons. Cada planta acostuma a tenir de 6 a 10 beines, segons la varietat de cigró[17]. La sembra de les cigroneres acostuma a fer-se formant rengleres deixant una separació entre renglera i renglera de 60 o 70 centímetres i plantant de 12 a 14 llavors cada metre, de manera que al final es troben cigroneres cada metre o metre i mig tenint en compte les que acaben germinant. Aquesta estructura ordenada del camp podria facilitar el modelat de l'entorn. Per tal de fer la recol·lecció dels fruits, un cop ja madurs, es pot fer de diverses maneres. Segons la varietat de cigró i el país d'origen el mètode de recol·lecció varia. Podem trobar que algunes de les varietats es recol·lecten a mà tallant la planta per la tija o bé arrencant-la del terra i deixant que s'assequi. Una altra opció molt utilitzada consisteix en utilitzar una collitadora modificada per tal de trillar la planta separant-ne el gra.



Figura 1: Cigronera on s'aprecien les beines on s'hi ubiquen els cigrons.

A banda de l'ésser humà hi ha diverses espècies animals molt interessades en assaborir aquest llegum. En concret, per les necessitats de l'estudi, s'han triat dues plagues molt destructives per al corneu del cigró. És el cas de l'*Heliothis Armigera* i de la família dels *Bruchidae*, d'entre els quals hi destaquem el *Bruchus Rufimanus*. Totes dues plagues representen un gran escull que frena la producció de cigrons malmetent les collites. En especial els de la darrera família, que perforen el fruit fent que no sigui apte per al consumidor.

L'*Heliothis Armigera* [7][1][9] és un lepidòpter que en la seva fase d'eruga es coneix també amb el nom d'eruga del tomàquet, del blat de moro i del cotó. Només amb aquesta descripció el lector pot fer-se una idea del possible impacte econòmic que representa aquest insecte en el comerç i encara es veu més clarament si s'esmenta la seva gran voracitat i les dificultats que apareixen a l'hora de controlar-la. Com s'observa, afecta en gran mesura a cultius d'elevat valor econòmic, i en el nostre cas també és una espècie perjudicial per a la cigronera.

Generalment prefereix el fruit a les fulles per alimentar-se i prefereix el blat de moro per realitzar la posta dels ous. És per això que alguna de les tècniques de control d'aquesta plaga consisteix en sembrar un camp "parany" de blat de moro i esperar que les erugues l'habitin; llavors es tallen els fruits i es donen de menjar al bestiar. Per ser efectiu, per cada 10 o 20 línies de cigronera, cal sembrar dues línies de blat de moro. Tot i que en un mateix fruit puguin habitar diverses erugues, mostren un patró de canibalisme molt acusat, el que fa que a la pràctica sigui difícil trobar més d'una eruga en el mateix fruit: les erugues grans es menjarien a les més petites.

En nombrosos països s'ha observat una gran resistència a plaguicides en l'*H. Armigera*, el que dificulta enormement el seu control. Fins i tot els darrers piretroides sintètics comencen a ser tolerables per aquests insectes. Els mètodes que acostumen a recomanar-se per al seu control són mètodes químics amb clorpirifs, endosulfan, metomil i peretroides o bé amb *Bacillus thuringiensis*. Però quan la eruga és ja gran ofereix una resistència important a aquests components. Com pot comprovar-se l'únic mètode que demostra ser efectiu consisteix en desapropiar part de la collita, perdent productivitat. Aquest treball esbrinarà si hi ha alguna altra forma d'aconseguir minimitzar les pèrdues.



Figura 2: Detall d'una eruga de l'espècie Heliothis Armigera.

Pel que fa a la família dels *Bruchidae* [7][16][20][2] o corcs de les lleguminoses, són uns animals d'una mida que oscil·la entre els 2 i els 8 mil·límetres, de cos oval i coloració fosca o negra. Són devoradors de llavors de lleguminoses en estat larvari i de pol·len i nèctar de flors en la seva forma adulta, que generalment no produeix danys.

El seu estat larvari es divideix en dues parts: una primera que va des de l'eclosió de l'ou fins que troba aliment i penetra en la llavor, i una segona en la que roman fins acabar la resta del seu desenvolupament alimentant-se al seu interior. En general s'acostuma a trobar tan sols una larva per llegum, ja que en el moment de la posta la femella explora la llavor i evita posar-hi els ous si hi detecta prèviament alguna larva. Per evitar aquesta plaga s'utilitzen mètodes preventius com poden ser la utilització de llavors sanes, que eviten plantes febles i de baix rendiment i que alliberarien l'insecte ubicat al seu interior.

El gènere *Bruchus* acostuma a combatre's al camp amb aplicacions cada 2 o 3 setmanes de malatió o triclofó. El seu impacte econòmic en el cultiu de les llegums és molt important, ja que les malmeten impossibilitant el seu consum humà. Els *Bruchidae* tendeixen a parasitar les plantes més desenvolupades i, com que les plantes més desenvolupades acostumen a trobar-se cap el sud, són aquestes les que comencen a atacar.

Alguns estudis[13] sobre espècies emparentades amb aquests coleòpters, del gènere dels *Curculionidae*, mostren una velocitat de

dispersió d'aquests insectes de $2.8 \pm 3.58 \text{ cm}^2/\text{h}$. Com que aquest no és un treball sobre biologia i es basa en estimacions, podem considerar que aquest factor de dispersió es similar per als corcs de les lleguminoses. En afegir aquestes dades al model, la simulació serà molt més precisa.



*Figura 3: Detall d'un Bruchus
Rufimanus adult.*

2.2 Tria d'un robot adequat

Abans de seguir amb el següent pas, que consisteix en triar un robot adequat per a l'extermini, caldrà establir quins seran els mètodes per eliminar els insectes. En alguns camps, per eliminar les erugues de l'*Heliothis Armigera* s'acostuma a cobrir el terra amb un plàstic o una tela. Llavors, es sacseja fortament la cigronera fent caure les erugues al terra. Finalment, es retiren aquestes teles o plàstics i d'aquesta manera també es retiren les erugues. El procediment que es durà a terme en el nostre cas serà molt similar.

2.2.1 Requeriments: formes d'eliminar insectes

El robot sacsejarà amb força la cigronera fent caure les erugues si n'hi hagués al terra. Una vegada allà, utilitzant el reconeixement de contorns i de textures per imatge[11] serà possible distingir una eruga de la resta del terra. Un cop localitzada l'eruga, mitjançant una cànula que el robot podrà estirar es ruixarà l'eruga amb una petita quantitat de pesticida. D'aquesta manera s'evita que la resta de pesticida contamine la planta.

El fet de poder determinar si un element de la imatge captada és una eruga o no, no és trivial i requereix unes condicions d'il·luminació òptimes i un ús intensiu de la capacitat de càlcul del processador. És per aquest motiu que s'ha triat una solució basada en el processador Atom, enlloc d'utilitzar un microcontrolador. Degut al bon rendiment que han estat demostrant aquest tipus de processadors i a la bona acollida que han tingut en centres dedicats a la visió per computador, resulta una opció molt encertada.

Pel que fa als corcs, els tractaments més habituals són principalment químics. Però el mètode que utilitzarà el robot serà lleugerament diferent. Una primera aproximació podria fer-se tenint en compte els desperfectes que presentaran les beines dels cigrons en haver-hi entrat els insectes. El robot, mitjançant reconeixement per imatge podria detectar-hi la presència dels insectes i eliminar d'alguna manera les beines ocupades pel corc. No obstant, aquest mètode presentaria problemes com la possible presència d'ombres que dificultarien la detecció de les beines i dels seus possibles forats i, el més greu de tots, que requeriria una gran capacitat de càlcul (necessària per fer el reconeixement d'imatges). Aquesta necessitat de càlcul intensiu revertiria en l'ús de les bateries que necessàriament han d'utilitzar aquests robots i en disminuirien l'autonomia. S'ha optat doncs per una altra solució més ràpida i fàcil.

En el cultiu de cereals també existeixen plagues de diferents tipus de corcs. Les lleis de molts països, com per exemple Canadà, fixen la tolerància gairebé a zero pel que fa al gra que no estigui sa, entenent per sa que no hi hagi cap insecte allotjat. És per això que s'han desenvolupat sistemes de detecció de corcs molt econòmics i ràpids. Alguns dels que s'han pogut veure involucren l'anàlisi de l'espectre de freqüències al voltant dels infrarojos[12], però aquest procés involucra un espectròmetre, un aparell força voluminós i que no resultaria apte per a un robot de baix cost.

Una altra solució comporta la detecció dels sons que efectuen els corcs dins el gra[5]. Alguns estudis sostenen que a distàncies d'alguns centímetres pot detectar-se la presència de corcs dins del gra amb l'ajut d'uns elements simples com micròfons, amplificadors i una bomba petita d'aquari. Per aproximadament uns 100\$ es pot aconseguir un sensor d'aquesta mena, el que el fa apte per un robot com el que es necessita. D'aquesta manera es pot detectar la presència de corcs a una planta a distància eliminant el soroll de fons i comparant la freqüència dels sons que es reben. Si es detecta la presència de corcs, s'aixafen les beines de la cigronera per evitar que la plaga s'escampi. D'aquesta manera, les larves del corc queden aixafades i no arribaran a l'estadi d'adults, evitant possibles futures plagues.

Per tots dos mètodes serà necessari disposar de dues càmeres que permetin al robot una visió estèreo. Sense ella seria molt complicat trobar les distàncies per accionar el braç i interactuar amb la planta i amb els insectes. Aquestes dues càmeres aniran muntades sobre el braç del robot, el que permetrà evitar en la majoria de casos les ombres que podrien fer difícil la detecció dels objectes.

2.2.2 Tria de components

Una vegada determinats els mètodes d'eliminació d'insectes, cal trobar el robot més adient. Resulta de vital importància centrar-se en les característiques de l'entorn per trobar la configuració més escaient. El terreny serà lleugerament accidentat, cosa que determina el mitjà de locomoció. S'han analitzat solucions basades en desplaçament mitjançant erugues i també un altre tipus de desplaçament, més convencional, basat en rodes.

Els avantatges del desplaçament amb erugues són bàsicament la facilitat d'evitar els obstacles passant-hi pel damunt i la simplicitat de manteniment. Malgrat això, els kits de desplaçament amb erugues que s'han estudiat permeten assolir velocitats relativament baixes, aproximadament d'1'7 Km/h. Segurament, seguint investigant, seria possible trobar solucions similars amb una velocitat superior, però de preu més elevat.

Ja descartades les erugues, queda l'altra possibilitat, el desplaçament amb rodes. Es tracta del mecanisme més utilitzat, i per tant, més econòmic. Presenta un clar avantatge pel que fa a la velocitat i a la simplicitat del control, per contra necessitem més manteniment, doncs cal mantenir les rodes inflades i hi ha més peces mòbils. Les arrels de les cigroneres no afecten al terreny en gran mesura, però sí que resulta interessant poder evitar petits obstacles i desnivells. És per això que s'ha optat per dotar de tracció a totes les rodes. A més, cal decidir quantes rodes ha de tenir. Havent estudiat solucions de quatre i sis rodes, trobem que les sis rodes resulten especialment útils quan al terreny s'hi poden trobar obstacles importants, i no és aquest el cas dels camps de cigrons. D'aquesta manera, s'estalvia la despesa energètica que suposaria haver de moure dos motors addicionals. Aquesta solució permet assolir una velocitat de 3'3 Km/h.

Per tal d'actuar amb les plantes i els insectes, el més recomanable serà disposar d'un braç articulat amb un *gripper* al final. Els requeriments d'aquest braç són força clars: ha de poder arribar a l'alçada màxima de

la cigronera i també ha de ser capaç de sacsejar-la. A més, ha de tenir una força suficient com per poder aixafar beines de cigrons encara verds.

Fins ara s'han analitzat els requeriments necessaris per poder actuar amb l'entorn, però encara cal triar la manera més adequada de rebre impulsos externs. Per tal d'analitzar la presència i la ubicació d'erugues s'utilitzaran dos sensors CCD muntats sobre el braç, que faran possible detectar la textura i el contorn d'una eruga comparant-la amb la del terra. Per detectar la presència de corcs, en canvi, s'utilitzarà un sensor basat en micròfons, que per mitjà del sons que emeten els corcs en menjar-se el cigró, detectaran les plantes que estiguin infestades. Per tal de desplaçar-se amb celeritat i evitant impactes, s'utilitzaran sensors de distància per ultrasons i sensors de distància per infrarojos. El posicionament del robot podria realitzar-se amb la inclusió de balises al camp, infraroges o bé de ràdio, que assenyalessin determinades distàncies.

Encara queda per determinar quin processador utilitzarà el robot. S'han estudiat alguns dels microprocessadors més habituals en aquesta mena de *kits* de robots, i han estat descartats per tenir massa poca capacitat de càlcul i no disposar d'elements per comunicar-se amb altres robots. És per aquest fet que s'ha optat per incloure una placa base mini-ITX amb un processador Intel Atom. D'aquesta manera, es podrà controlar tant el reconeixement d'imatges, com el moviment i la comunicació d'una manera senzilla, permetent alhora la reconfigurabilitat del robot. Les dades i el programari necessaris pel funcionament del robot podran emmagatzemar-se en una memòria flash convencional.

El fet d'incorporar una arquitectura de PC podria influir en el consum, però de seguida es veurà que el consum no és més elevat que el necessari pel desplaçament del robot i pel moviment del braç.

Els components escollits finalment són:

Base A4WD v2

ALD5 Robotic Arm Combo Kit

Wrist Rotate Upgrade (Heavy Duty)

Dues Càmeres CCD

Placa Base D945GCLF2 i Processador Atom

Sensor Ping))) Ultrasonic Distance Sensor

Sensor IR GP2D12

2.3 Consum energètic

Els motors i servos inclosos en alguns dels *kits* de Lynxmotion es detallen en el càlcul de consum. Tenint en compte els elements triats, un consum estimat del robot seria el següent:

Desplaçament del robot:

4 Motors GHM-1: $4 * 233 \text{ mA} * 12 \text{ V} = 11.18 \text{ W}$

Controlador servos SSC-32 : $250 \text{ mA} * 6\text{V} = 1.5 \text{ W}$

Sensor GP2D12: $50 \text{ mA} * 5 \text{ V} = 0.25 \text{ W}$

Sensor Ping))) Ultrasonic Distance Sensor: $35 \text{ mA} * 5 \text{ V} = 0.175 \text{ W}$

Placa base+processador Atom D945GCLF2: 30 W

Total: $11.18+1.5+0.25+0.175+30 = 43.105 \text{ W}$

Moviment del braç (amb una càrrega mitja):

1 Servo mida estàndard HS-422: $500 \text{ mA} * 6 \text{ V} = 3 \text{ W}$

2 Servos mida estàndard HS-475HB: $2 * 600 \text{ mA} * 6 \text{ V} = 14.4 \text{ W}$

1 Servo mida estàndard HS-645MG: $1000 \text{ mA} * 6 \text{ V} = 6 \text{ W}$

2 Servos gran escala HS-5745MG: $2 * 2800 \text{ mA} * 6\text{V} = 33.6 \text{ W}$

Sensor GP2D12: $50 \text{ mA} * 5 \text{ V} = 0.25 \text{ W}$

Sensor Ping))) Ultrasonic Distance Sensor: $35 \text{ mA} * 5 \text{ V} = 0.175 \text{ W}$

Placa base+processador Atom: 40 W

Total: $3+14.4+6+33.6+0.25+0.175+40 = 97.425 \text{ W}$

Captura i processament d'imatges:

Càmeres CCD: $2*50 \text{ mW} = 100 \text{ mW}$

Sensor GP2D12: $50 \text{ mA} * 5\text{V} = 0.25 \text{ W} = 0.25 \text{ W}$

Placa base+processador Atom: 50 W

Total: $50+0.25+0.1 = 50.35 \text{ W}$

Comunicació robot:

Placa base+processador Atom + WiFi: **41 W**

Total: 41 W

Aquests consums estimats són molt aproximats. Els consums referents al moviment del braç s'han estimat en funció del consum sense càrrega i el consum màxim, agafant un punt intermedi. Pel que fa al consum de la placa base, s'han determinat els consums en base a la quantitat de càlcul i de hardware necessari per dur a terme les diferents tasques.

Caldria tenir en compte que no tots els motors del braç es mouen alhora, pel que s'ha cregut convenient fixar el consum aproximadament en un 75% del que s'ha calculat. Això significaria un consum d'uns 75 W aproximadament.

2.3.1 Temps necessari per a cada tasca

En cas de detectar una planta en mal estat, el robot s'hi apropa per veure'n la causa, és a dir, si és per culpa dels corcs, de les erugues o bé de tots dos. Aquesta detecció podria portar-se a terme mitjançant reconeixement per imatge de la planta, comprovant-ne l'alçada, el color o bé l'abundància de fulles o de beines.

Per detectar les erugues serà necessari sacsejar la planta durant aproximadament uns 20 segons i posteriorment analitzar el terra al voltant de la planta cercant les erugues. Fixant-nos un sector d'1 m² dividirem el sector en 4 fragments més petits, de 625 cm² equivalent a sectors de 25*25 centímetres. Sabent que segons les dades experimentals que s'han obtingut amb un processador Pentium III a 750 Mhz, es triga aproximadament dos minuts en processar una imatge de 108*176 píxels, podem extrapolar que si assignem 140*140 píxels a aquests fragments amb un processador Atom a 1'6 Ghz com el que té el robot d'aquest treball assumirem que trigarà 180 segons en fer l'anàlisi d'aquest perímetre. A més, per eliminar els insectes, es trigarà 60 segons per cadascuna de les erugues a eliminar, tenint en compte que s'han de localitzar i ruixar amb insecticida.

En el cas dels corcs, es disposa de menys informació pel que fa a la durada dels algorismes. De qualsevol manera es pot pensar que capturant durant 10 segons el so de cadascuna de les beines de la cigronera, es pot detectar la presència de corcs en una planta. Tenint en compte que cada planta té de mitjana 7.5 beines, això equival aproximadament a uns 80 segons de detecció. Si a més cal sumar-hi el

temps de detectar la posició de les beines, aquest temps s'incrementa fins arribar a uns 180 segons. Després d'aquesta detecció, cal esclafar les beines infestades, pel que es requereixen 45 segons addicionals per cada beina.

2.3.2 Consum energètic necessari per a cada tasca

Per tal de reconfigurar-se, cada robot que passi de matar corcs a matar erugues haurà d'extreure la cànula mitjançant la qual procedirà a matar-les. És per això que s'ha considerat un temps d'uns 60 segons per extreure la cànula i uns 45 segons per tornar-la a desar, en el cas que es tornin a exterminar corcs. Cal tenir en compte, que a més de reconfigurar-se, el robot també haurà de repetir el procés d'inspecció de la planta en busca d'erugues o corcs, ja que el procés per localitzar-los varia en funció de l'espècie. És per això que la reconfigurabilitat té un cost molt elevat i cal establir una política adequada per reduir el consum d'energia i, per tant, el temps que els robots poden treballar al camp.

Fixant-nos ara en la quantitat d'energia consumida pel robot, pel desplaçament hi haurà una despesa d'uns 43'105 W com ja s'ha vist. Per tal d'eliminar erugues, s'han de moure els braços i utilitzar intensament el processador, d'aquesta manera s'ha determinat que es consumiran 117'425 W. L'eliminació dels corcs implica una menor capacitat de càlcul, però cal també moure el braç articulat i utilitzar el sensor de sons. És per aquesta raó que s'ha estimat el consum en eliminar corcs en uns 80'425 W. Per tal de reconfigurar-se passant de matar corcs a matar erugues, cal desplegar la cànula, fent una força ascendent, pel que considerarem que aproximadament es fa un treball equivalent a 80 W. Si el treball és al contrari, desar la cànula per passar de matar erugues a matar corcs, es farà un treball menor ja que no s'ha de vèncer la força de la gravetat. No obstant, no es pot deixar el braç articulat lliure, pel que es considera una despesa energètica de 75 W.

Per emetre la seva configuració, un robot empraria 41 W, tot i que amb les estratègies utilitzades no ha estat necessari enviar la pròpia posició ni el color dels punts que tracta. No obstant, si en un futur s'afegís una altra estratègia amb qualsevol d'aquestes funcions, caldria tenir en compte aquesta dada. El radi de comunicació del robot podria ser d'uns 100 metres, ja que el robot disposa d'una tarja WiFi 802,11g per tal de dur a terme la comunicació entre robots.

El robot funcionarà amb bateries. Tenint en compte que els portàtils actuals funcionen amb bateries de 4400 mAh a uns 14.4 V, no resulta estrany pensar que un robot d'aquesta mena podria portar dues bateries

com aquestes. D'aquesta manera es podria aconseguir una font d'energia capaç de subministrar 65.12 Wh, o el que és el mateix, 469 KJ, ja que el Joule és la unitat amb la que treballa el simulador.

Una vegada definits aquests paràmetres no funcionals, es podrà avaluar el resultat de qualsevol de les estratègies emprades. Però abans, cal completar l'eina que servirà de suport per avaluar-les, el simulador.

Capítol 3

Simulació i Modelat de l'Entorn

Un simulador és una eina que permet reproduir el comportament d'un sistema sota determinades circumstàncies. El fet d'utilitzar un simulador, en comptes de comprovar empíricament cada petit canvi permet fer multitud de proves amb tots els comportaments que s'implementin d'una forma fàcil i ràpida. Si no fos gràcies a una eina com aquesta, caldria programar un bon grapat de robots i observar-ne el comportament en un entorn real i, el que és més important, en temps real. Això comportaria uns costos que no es poden assumir en aquest estudi i no aportaria diferències substancials en els resultats. És per això que s'ha considerat la opció més adequada en un cas com el que ens ocupa. Aquest capítol detalla com es fa aquest simulador i quins són els aspectes importants a tenir en compte a l'hora de dissenyar-lo.

3.1 Simulador original: R&B Sim

Per tal de dur a terme l'estudi dels comportaments socials dels robots una de les coses essencials consisteix en disposar d'una eina que ens permeti avaluar l'efectivitat d'aquests comportaments. És per això que serà necessari elaborar un simulador, però no s'ha partit de zero, sinó que s'ha agafat un simulador existent anomenat R&B Sim, que va ser un projecte final de carrera d'en Daniel Alcantarilla.

Aquest simulador estava plantejat per observar comportaments d'eixams de microrobots. És per això que caldrà fer algunes modificacions a aquest programa que s'aniran detallant a continuació per tal que s'adapti al problema sobre el que s'ha centrat el treball.

El simulador R&B Sim, simula el comportament d'uns robots mentre resolen un problema de joguina anomenat Red & Blue, en el que disposem d'un terreny on hi ha dispersats alguns punts de colors blau i vermell. Es disposa d'una sèrie de robots reconfigurables que poden ser de color blau o vermell i que poden interactuar amb els punts del seu mateix color en tocar-los. Poden detectar els obstacles tocant-los i llavors han d'examinar de quin color es tracta. Si és del seu mateix color, hi efectuen les operacions que calgui i els desactiven transformant-los en

punts negres. Si no és del seu mateix color, poden reconfigurar-se canviant el robot de color, sempre i quant hi hagi un altre robot de l'altre color a una distància prou petita. L'objectiu del joc és que tots els punts quedin de color negre.

Ara bé, el simulador serveix per robots molt simples amb un comportament molt primitiu. Per a la classe d'estudi necessària per aquest treball és imprescindible modificar el simulador per tal de permetre incloure comportaments més sofisticats. Si s'observa el graf d'estats dels microrobots, es comprovarà de què estem parlant de manera més senzilla.

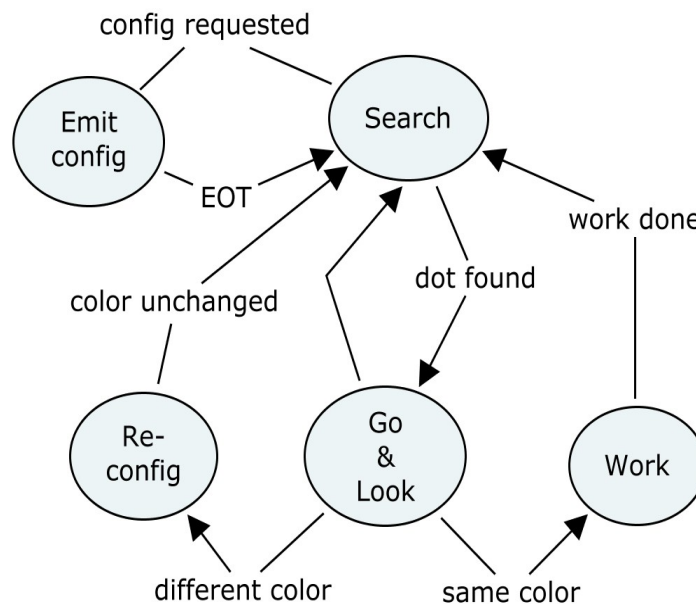


Figura 4: Diagrama de la màquina d'estats que regeix el funcionament dels robots al R&B Sim.

Així, si el que es pretén es modificar el comportament dels robots, caldrà modificar les estratègies de cerca i reconfiguració. D'aquesta manera, creant-ne de noves i avaluant la velocitat en trobar una solució i el consum energètic necessari per arribar-hi, podrem ser capaços de comparar aquests comportaments.

Al simulador R&B Sim original, la cerca d'un nou punt és purament aleatòria, aconseguint un moviment brownià dels robots. No obstant, el cas d'un camp de cigrons és sensiblement diferent. Si es planten les cigroneres seguint estructures regulars serà possible determinar el següent punt a buscar sense haver de recórrer aleatòriament tots els punts. A més en el simulador original, la política de reconfiguració pot establir-se en dos valors: o sempre es reconfigura o mai ho fa. En els robots d'aquest estudi caldrà establir polítiques de reconfiguració més

elaborades basant-nos en la posició del robot (quant més al sud, més possibilitat de trobar-hi corcs i erugues, ja que són les plantes més desenvolupades) i en l'històric de plagues trobades a les darreres plantes.

Aquests dos factors influiran d'una forma considerable en el bon funcionament dels robots per a realitzar la tasca encomanada. En principi semblaria lògic pensar que quant més elaborades siguin aquestes estratègies més bon resultat s'obtindrà, però potser el que sembla més obvi no és pas tan real.

Per a fer les simulacions és necessari adaptar el simulador existent afegint-li un seguit de funcionalitats que resultin útils a l'hora de modelar el problema i d'obtenir-ne els resultats. A l'apartat següent es detallen les necessitats específiques de les simulacions als camps de cigroneres.

3.2 Modificacions del simulador R&B Sim

Per començar a adaptar el simulador existent a un simulador de camps de cigroneres, cal en primer lloc generar un camp de cigroneres amb les seves plantes i els seus insectes. Al simulador R&B Sim els robots s'ubiquen en un territori quadrat o rectangular on s'hi ubiquen els punts a tractar. En un camp de cigroneres, a més, hi haurà un seguit de plantes on s'hi posaran els punts. Cal, doncs col·locar dins el mapa aquestes plantes i fer que els robots siguin capaços de detectar-les.

3.2.1 Modelat de l'entorn

Per introduir les plantes a l'escenari s'han plantejat diverses solucions. En principi, com ja s'ha explicat anteriorment, en un camp de cigroneres les plantes es col·loquen formant rases, de forma que hi hauria uns carrils delimitats pels robots. El problema que sorgeix, és que aquestes rases fan complicat que més d'un robot hi circuli i més si tenim en compte que una de les estratègies de cerca a utilitzar serà la que ja existeix a l'aplicació actual: una estratègia aleatòria. És per això que s'ha optat per fer deixar certes zones de cada rasa sense dues plantes. D'aquesta manera s'obren camins per tal que els robots puguin canviar fàcilment de renglera. Un exemple de mapa de dimensions de 4 hectàrees, l'equivalent a 40.000 m² seria el següent:

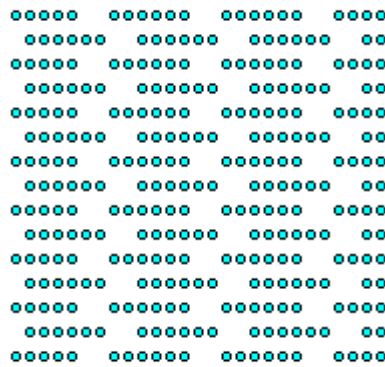


Figura 5: Representació d'un fragment de 200m x 200 m d'un camp de cigroneres.

A més, serà necessari ubicar inicialment els insectes. Considerarem que aquests no es mouen, ja que les erugues no canvien ràpidament de planta i els corcs es troben a l'interior de les beines. A més, la velocitat de dispersió dels corcs és molt limitada, per tant, no resulta massa equivocat suposar un comportament estàtic. D'aquesta manera es mantindran immòbils al llarg de tota la simulació, cosa que facilitarà la ubicació dels punts.

Per tal de situar els insectes a l'inici cal tenir en compte una sèrie de casuístiques. En primer lloc, hom podria trobar-se a l'inici d'una plaga. En aquest cas els insectes s'ubicarien a un sector limitat del terreny. Poc a poc els insectes es van estenent des de les vores del camp fins la part central. En una darrera instància, els insectes poden trobar-se repartits per tot el terreny de forma més o menys aleatòria. Molts estudis confirmen el que també resulta de sentit comú, els insectes prefereixen les plantes més fortes i sanes. Com que les plantes del sud són les més afavorides per la llum solar, aquestes seran les que compleixin les característiques esmentades. És per aquest fet que s'haurà de primar la ubicació dels insectes cap a l'extrem sud.

A més de tot el que s'ha exposat, també és necessari adaptar un altre aspecte de la distribució inicial dels insectes. Al R&B Sim, els punts es col·locaven en posicions aleatòries de l'escenari. Però els insectes tenen un comportament força més capritxós i prefereixen viure sobre les plantes, per tenir l'aliment més a la vora. És per això que ha estat necessari col·locar els insectes al perímetre de les plantes.

S'ha vist que depenent de la fase de la invasió caldrà posicionar els punts d'una manera o d'una altra. Per això s'han inclòs diverses alternatives diferents com a situació inicial pels punts. És possible

col·locar els punts aleatòriament al mapa, però també és possible fer-ho segons una normal, de manera que quedarien al centre de l'escenari la majoria dels punts, i en menys abundància a les vores; no obstant, s'ha primat l'extrem sud en detriment del nord per tal de beneficiar les posicions més assolables. També s'ha inclòs la possibilitat d'ubicar els punts a les vores del mapa. D'aquesta manera els punts es col·loquen seguint una funció de distribució de probabilitats similar a la inversa d'una normal. També en aquest cas serà més fàcil trobar punts a l'extrem sud i en trobarem més a les vores que no pas al centre. Això no vol dir que al centre no n'hi hagi, però l'afluència d'insectes serà sensiblement menor.

De la mateixa manera en la que es distribueixen els insectes, podem fer la distribució dels robots. Això vol dir que els podem deixar aleatòriament a alguns punts del camp o podem primar on s'acostumen a trobar els insectes segons la fase de la plaga. Podríem pensar que si els insectes es troben aleatòriament pel mapa, pot resultar útil col·locar els robots de la mateixa manera. En canvi, si es troben al centre o a les vores, resultarà més eficaç col·locar-los seguint una funció de distribució normal amb centre al centre del camp i desplaçada lleugerament cap al sud o bé si es troben a les vores, ubicar a les vores també els robots. Així s'evita en gran mesura el desplaçament excessiu i innecessari dels robots, que no tenen recursos energètics il·limitats i que, d'aquesta manera, veuen incrementat el seu temps de treball útil.

3.2.2 Modelat del comportament dels robots: estratègies

Al nostre simulador sense modificacions, els robots segueixen un diagrama d'estats senzill, que ha calgut mantenir en la mesura del possible. No obstant, aquest problema de joguina no contemplava la possibilitat de que els robots es veiessin obstaculitzats per alguns objectes, com és el cas de les plantes. És per això que s'han afegit col·lisions i les estratègies necessàries per buscar altres objectius quan el camí cap a un d'ells es veu interromput.

També resulta un factor clau clau poder establir diferents polítiques de cerca dels punts. En principi, sembla raonable pensar que una estratègia diferent a la cerca aleatòria dels insectes per totes les plantes. Si es coneix que en alguns estats de la invasió les plantes de les vores són les més atacades, sembla raonable pensar que aquests haurien de ser els punts més investigats pels robots. D'altra banda, ja s'ha esmentat que aquesta situació es dona només en algunes fases de la invasió, pel que serà apropiat afegir d'altres polítiques de cerca diferents.

Disposem, fins ara, d'una cerca totalment aleatòria i una altra enfocada a les vores, propiciant la cerca a les zones del sud, d'una forma similar a l'estratègia d'ubicació inicial dels punts. També resulta útil incloure una cerca que segueixi una funció de distribució normal desplaçada cap al sud, que resultarà útil quan es divideix el camp en fragments.

Les estratègies de cerca que s'han esmentat són força genèriques, però també se n'ha inclòs alguna que es veu alterada pels punts que s'han trobat fins el moment actual. Utilitzant un paràmetre configurable, els robots són capaços de desar en memòria la posició d'un nombre d'insectes trobats determinat, així com la seva posició. A partir d'aquestes dades, i calculant-ne una normal amb una desviació fruit de dividir per una constant la mida del camp, es poden obtenir punts propers on ja s'han trobat insectes prèviament. D'aquesta manera, intuïtivament es pot percebre que si s'han trobat alguns insectes en una zona, és molt probable que a prop n'hi hagi més. Aquesta estratègia presenta la limitació que els insectes més aïllats, tenen menys probabilitat de ser tractats. A més, com que és necessari disposar d'algun punt a partir del qual trobar-ne d'altres, cal començar utilitzant una altra estratègia, que en el nostre cas consisteix en buscar punts de forma aleatòria.

Totes aquestes estratègies de cerca que s'han explicat es basen en trobar punts on habitualment es situen els insectes o on se n'han trobat, però no garanteixen uns bons resultats. Cal tenir en compte que les distàncies que recorren els robots i la varietat de punts que podrien trobar-se en la seva cerca poden ser molt extenses. També resulta de vital importància considerar les col·lisions entre robots i el fet de tractar punts que d'altres robots ja han tractat. Veiem-ho més clar amb un exemple: dos robots que tracten punts del mateix color estan tractant les plantes d'una mateixa rasa i es van apropant. Arribarà un punt en que ambdós robots xocaran i, en cas de ser capaços de mantenir la seva trajectòria es trobarien amb plantes prèviament tractades per l'altre robot, per tant, cercar entre aquelles plantes un punt del seu mateix color resultaria inútil. Per evitar situacions similars, que farien incrementar la despesa energètica i el temps necessari per a tractar el problema, s'ha establert un altre tipus de cerca, que anomenarem A-Search.

La cerca A-Search és una cerca més avançada que l'anterior. De la mateixa manera que a l'hora de circular per les carreteres els éssers humans complim una sèrie de normes que ajuden a evitar xocs, a millorar la fluïdesa del trànsit i, en definitiva, a garantir l'èxit del desplaçament, la idea bàsica de l'A-Search es basa en la mateixa filosofia. Si establim unes normes pel que fa a moviment, és a dir la cerca, i reconfiguració dels robots, podem obtenir millors resultats, o com

a mínim és el que semblaria raonable. Per tal de corroborar-ho, serà necessari comprovar-ho mitjançant una simulació.

Aquesta estratègia consisteix en que tots els robots es desplaçaran en un sentit determinat que seran capaços d'identificar. Poden moure's a est, oest, nord i sud i cercaran sempre les plantes de la seva dreta, ignorant les de l'esquerra. D'aquesta manera si dos robots passen per la mateixa rasa en sentits oposats, tot i que siguin del mateix color no es produirà una situació en la que ambdós robots investiguin sobre plantes ja desinfectades, ja que un treballarà sobre les plantes d'un costat i l'altre, sobre les del costat restant. A més, d'aquesta manera es redueixen el nombre de col·lisions entre robots. Si per casualitat un robot quedés encallat amb una planta o amb un altre robot canviarà la direcció del desplaçament. Així, si el robot es desplaça cap a l'est i s'encalla, es desplaçarà cap al sud, si es torna a encallar, cap al nord, i si succeeix el mateix, cap a l'oest i així successivament seguint el sentit de les agulles del rellotge. D'aquesta manera, a la seva dreta sempre es procura que hi quedin les plantes, per tal de minimitzar el temps que el robot no detecta cap insecte. El fet d'incorporar detecció de robots a la mateixa rasa que comparteixin el color i es desplacin en el mateix sentit podria haver ajudat a fer més eficaç l'algorisme de desplaçament, però alhora hauria complicat de forma considerable l'elaboració de l'estratègia. A més, els robots triguen un temps força elevat en detectar i tractar les plantes, pel que tampoc resulta gens descabellat fer que diversos robots del mateix color circulin per una rasa en el mateix sentit. Fins ara s'ha parlat de l'estratègia que seguiran els robots, però a l'hora d'implementar-ho dins el simulador, per les pròpies limitacions inherents al R&B Sim i ja que es pretenia conservar-ne l'estructura al màxim, és força diferent. Així, cada robot per tal de determinar el següent punt on es desplaçarà, primer calcularà la distància fins a cadascuna de les plantes del camp, i escollirà la més propera que pugui trobar desplaçant-se en la seva direcció i es trobi a la seva dreta. Si allà detecta que la planta presenta un aspecte feble i és susceptible d'estar infestada, procedirà a la detecció d'insectes dels que actualment tracta el robot. Depenent de la política de reconfiguració, tractarà la planta o procedirà a cercar el següent punt.

Un lector hàbil podria haver-se adonat que ja hi ha algorismes de cerca que minimitzen les distàncies i asseguren una millor cobertura del terreny. No obstant, les limitacions que presenta aquest simulador s'han fet notar durant tota la fase de desenvolupament d'aquesta solució. En realitat l'R&B Sim és un simulador basat en un altre simulador anomenat MuRoS, que al seu torn està basat en un simulador de partícules. Això vol dir que la física està perfectament modelada, i això té un cost en el càlcul força important. És per això que algorismes d'intel·ligència artificial per trobar camins mínims, com l'A*, han quedat ràpidament descartats

degut a que l'estratègia A-Search ja resulta molt lenta de simular en un temps raonable, entenent com a raonable que es puguin dur a terme les simulacions necessàries en les 450 hores que ocupa un projecte final de carrera.

Fins ara s'ha parlat de la ubicació dels insectes, dels robots i de les estratègies de cerca a utilitzar. Encara ens queden per exposar totes aquelles estratègies referents a la reconfiguració del robot, que permetin que passi de matar una sèrie d'insectes, a una altra.

En arribar a una planta que cal inspeccionar, on hi podria haver algun insecte es procedirà a analitzar-la segons el tipus d'insecte que tracti el robot. Si la planta presenta signes d'estar infestada per la mateixa plaga que el robot està atacant, es procedirà a l'eliminació de l'insecte, però en cas contrari serà necessari o bé reconfigurar-se per tractar l'altra plaga o bé deixar que un altre robot s'ocupi de solucionar el problema. S'ha de tenir en compte que la reconfiguració implica tornar a inspeccionar la planta i preparar les eines necessàries per a l'eliminació dels insectes, cosa que implica un elevat cost, que no sempre convidrà assumir.

Originalment el simulador permetia dues formes de reconfiguració, que impliquen que un robot es reconfiguri sempre que arribi a un punt d'un color diferent al seu, o bé que no ho faci mai. Totes dues polítiques poden tenir resultats diferents segons els problemes a resoldre. En el cas de les cigroneres, pot dependre en gran mesura del robot i de l'estratègia de cerca emprada. De qualsevol manera, s'ha cregut convenient afegir una altra política de reconfiguració, que consistirà en avaluar els darrers punts trobats. En el cas d'haver trobat menys d'un cert nombre de punts d'un color, tot i detectar que en una planta hi ha presència dels insectes que no se sap tractar, el robot no es reconfigura. Així, tan sols havent fracassat en un nombre concret d'ocasions en la cerca dels insectes a matar, es procedeix a la reconfiguració. D'aquesta manera es limita en gran mesura el cost que representa haver d'analitzar de nou la planta i matar l'altre tipus d'insecte. Depenent del nombre triat com a paràmetre, el resultat serà més o menys satisfactori. Si es tria com a valor el 0, el resultat serà que cada nou punt que trobi el robot de color diferent a l'actual, efectuarà la reconfiguració.

Amb tot aquest seguit de polítiques i opcions es poden simular les diferents situacions en les que podem trobar l'entorn. Però si no acotéssim la investigació d'alguna manera es podrien variar els paràmetres a la babalà i no s'aconseguiria extreure conclusions fiables sobre els comportaments socials, que no cal perdre de vista que, més enllà del control de plagues, és l'objectiu d'aquest estudi.

És per això que resulta essencial limitar l'entorn a una situació concreta. En aquest cas ens trobarem davant d'una de les situacions que ha presentat més problemes a l'hora de solucionar-ho en les diferents proves fetes amb el simulador. Es tracta d'una de les primeres fases de la invasió, en la que els punts es troben situats a les vores del camp. Per tal de minimitzar els desplaçaments inicials, també hi ubicarem els robots a les vores.

Els paràmetres que modificarem seran l'estratègia de cerca i l'estratègia de reconfiguració. Deixant fixe els nombres de robots i d'insectes d'un i de l'altre color, així com la mida del camp. Per facilitar la simulació, el camp tindrà una mida de 200 metres de longitud per 200 metres d'amplada. S'ha comprovat que ampliant aquesta mida a 500 metres per 500 metres el rendiment del simulador disminueix de forma acusada. De qualsevol manera, també s'utilitza un nombre reduït de robots, uns 25 per cada tipus d'insecte. Pel que fa al nombre d'insectes que trobem dins el camp, s'ha considerat una infestació que faci necessària una intervenció urgent i que mereixi la despesa d'ubicar-hi els robots. Es per aquest motiu que s'ha optat per situar-hi 100 corcs i 100 erugues. Si tenim en compte que segons la configuració del mapa que s'ha descrit en un escenari com l'actual hi ha més de 300 plantes, estem parlant que de ben segur més de la meitat estan infestades, cosa que justifica l'actuació.

Per tal de desar els resultats s'ha implementat una funció encarregada d'obtenir els valors d'energia consumida, de punts treballats de cada color i de nombre de reconfiguracions quan passen un nombre determinat de minuts. Per a les següents simulacions s'ha cregut convenient que aquest temps sigui de 15 minuts. Estimant la durada de les bateries i els consums del robot, una simulació hauria de simular un espai de temps d'entre 3 i 5 hores. D'aquesta manera es disposarà d'un nombre suficient de dades per treure'n algunes conclusions.

Capítol 4

Codificació de les adaptacions

El simulador R&B Sim està programat en Microsoft Visual C++. per tant, aquest ha estat el llenguatge de programació escollit per fer-hi les modificacions. Ha estat necessari fer 4 tipus de modificacions diferents, tot i que algunes d'elles estan molt relacionades. Les accions concretes per adaptar cadascuna de les funcionalitats del R&B Sim per a convertir-lo en ChickPeaSim es detallen tot seguit.

4.1 Adaptacions al modelat inicial

Ja s'ha comentat que era necessari incloure la presència de plantes i d'insectes al mapa. A més també és necessari afegir-hi diferents distribucions per als robots. Per tal de fer-ho s'ha creat un seguit de variables que l'usuari ajusta mitjançant un entorn gràfic i, d'aquesta manera, queda enregistrada la distribució inicial de punts i robots entre les opcions que es detallen. Per veure-ho d'una forma més intuïtiva, es pot observar la pantalla que veu l'usuari del simulador.

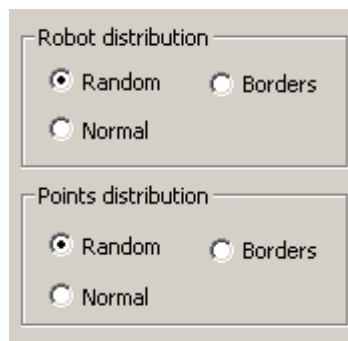


Figura 6: Opcions del simulador per a la ubicació inicial de punts i robots.

Per programar aquest seguit d'opcions s'ha modificat la classe `CsimulatorDoc`, que conté el mètode `OnParametrosEscenario()`, encarregat de carregar els valors de la finestra d'opcions per a l'escenari, anomenada `DlgEscenario` i s'encarrega de la inicialització de l'escenari.

4.1.1 Distribució inicial de les plantes

En primer lloc, cal afegir la manera de que a l'escenari hi hagi les cigroneres. Com ja s'ha explicat, hi haurà forats de mida equivalent a dues plantes cada cert nombre de metres per cadascuna de les rases formant vies per tal que els robots puguin passar entre elles. A les rengleres parells, estaran a unes posicions i a les imparells a unes altres.

Això facilitarà que en el cas de cerques aleatòries o probabilístiques els robots puguin canviar de rasa sense anar-se'n massa lluny. El codi que s'ha afegit al mètode és molt senzill i es presenta tot seguit:

```

/*****
/* Crear i ubicar plantes de l'escenari
*****/
for (int i=1; i<dlgEscenario.m_sizeX/12; i++ ) {
    for (int j=1; j<dlgEscenario.m_sizeY/7; j++ ) {

        if ((i%2==0&&(j%8!=0&&j%8!=1))|| (i%2==1&&(j%8!=6&&j%8!=7))) {
            //afegir una planta de 5*5 cada 12 unitats de mesura en horitzontal i 7 en vertical
            CSimulatorDoc::AddObstacle(new
                CObstacleCircle(CRect(j*7,i*12+5,j*7+5,i*12)));
        }
    }
}

```

Figura 7: Descripció del fragment de codi que situa les plantes a l'escenari seguint el patró descrit.

Fent una breu descripció del codi, les files parells estaran partides per forats a les posicions 6, 14, 22, ... cada 8 plantes, ocupant un espai equivalent a dues plantes. En canvi les rengleres senars, tindran forats a les posicions 0, 8, 16, 24, ... també cada 8 plantes. Aquest codi permet obtenir un disseny del mapa que ja s'ha mostrat en capítols anteriors.

4.1.2 Distribució inicial dels robots

Per ubicar els insectes i els robots s'ha ideat una funció anomenada GenerarXY(), que rep la distribució inicial dels insectes o dels robots i retorna uns punts de manera que aquests punts retornats segueixen la distribució seleccionada. Més endavant es detallarà aquesta funció, entre tant, pot observar-se la manera d'inicialitzar l'escenari tant pels robots de color vermell com pels de color blau.

Per detallar el funcionament del mètode GenerarXY() cal entendre primer les diferents possibilitats de distribucions inicials. Tal com s'ha explicat existeixen les opcions d'ubicar els objectes de forma totalment aleatòria, de fer-ho segons una normal desplaçada cap al sud o bé de fer-ho a les vores.

Per tal de triar una distribució aleatòria, la funció GenerarXY() retorna dos punts aleatoris compresos entre 0 i la dimensió del camp, tant per a l'amplada com per a la longitud. D'aquesta manera, s'obtenen valors vàlids per a situar-hi els objectes.

Si en canvi el que es pretén és situar els objectes seguint una normal, el procés comença amb la creació d'un generador de nombres que segueix una funció de distribució de probabilitats normal. En realitat hi ha dos generadors de nombres diferents, un per a cada eix. El generador de nombres aleatoris de l'eix corresponent a l'amplada, es centra a l'1, mentre que el de l'eix corresponent a la longitud es centra a l'1.25. D'aquesta manera hi ha una certa tendència a col·locar els objectes més cap al sud, que és el que succeeix als insectes a la natura. La desviació d'aquest generador de nombres és equivalent a una vuitena part de les dimensions del camp, que experimentalment han demostrat ser uns valors acceptables. A mesura que es van fent crides a la funció `GetNumber()` del generador, s'obtenen les coordenades necessàries per a la ubicació de robots i insectes.

Finalment, el cas més complexe és el de situar els insectes a la vora, que en realitat seria el més habitual en la natura. Per tal de fer-ho s'ha hagut de simular una funció amb una distribució de probabilitats semblant a la inversa d'una normal. S'han creat, de la mateixa manera que en el cas anterior dos generadors de nombres aleatoris seguint una funció de distribució de probabilitats normal. De la mateixa manera, també el generador de les coordenades corresponents a la longitud està desplaçat cap al sud. En aquest cas, però la desviació serà de 1. Amb aquests generadors s'obtenen valors que oscil·len habitualment entre 0 i 2, i amb menys freqüència s'obtenen nombres menors que 0 o majors que 2. Com que el cas més habitual, segons una normal seria obtenir un 1 o un nombre proper a aquest i el que ens interessa és que aquests punts corresponguin a les vores, es fa una conversió, de manera que si s'obté un 1, es trii una de les vores de forma aleatòria, si s'obté un nombre major que 2 o menor a 0, es tria un punt central. En canvi, si es tria un nombre entre 0 i 1, o entre 1 i 2, s'obtindrà un punt intermedi, més apropat a la vora quant més proper sigui a l'1.

El codi d'aquest generador de nombres és extens i seria feixuc col·locarlo en aquesta memòria, però el funcionament ha quedat descrit amb precisió. Amb totes aquestes opcions, el generador es comporta de la forma esperada segons la funció de distribució de probabilitats i els generadors de nombres aleatoris triats. Si es pretengués afegir altres distribucions inicials podria fer-se sense massa complicació afegint el codi necessari i els generadors adequats.


```

/*****
/* Crear i ubicar els robots blaus
*****/
for(i=0; i<dlgEscenario.m_robotblue; i++) {
    CString s;
    s.Format("%4d", i);
    GenerarXY(m_distribucio_robots,&NormalGenerator1X,&NormalGenerator1Y,
    |         &NormalGenerator2X,&NormalGenerator2Y, &rand_x,&rand_y);
    p = CPoint( rand_x, rand_y );
    robot = new CRobotForage(p.x, p.y, 0, WANDER,
    |         (short) m_robots.GetSize(), s, (short) 1, (short)1);
    robot->m_localMap = new CMapPath(m_globalMap, FALSE);
    AddRobot(robot);
    CRobotForage::AddRobotColor(1);
}

```

Figura 8: Fragment de codi per a crear i col·locar els robots de color blau al mapa.

4.1.3 Distribució inicial dels insectes

La distribució inicial dels insectes presenta una lleugera diferència respecte la ubicació dels robots. Cal que els punts estiguin situats a les plantes. Per aconseguir-ho, s'ha seguit el següent patró: primer es troba un punt que segueixi la funció de distribució necessària i posteriorment, es troba la planta més propera a aquest punt. Les distàncies, en aquest cas es calculen fins al centre de la planta. Finalment, aleatòriament s'obté un angle, que, fent les operacions pertinents es transforma a un punt de la vora de la planta. D'aquesta manera cadascun dels insectes creats pel programa es col·loquen a la vora d'una planta. Aquesta operació de trobar el punt corresponent a la vora de la planta més propera la duu a terme la funció `EncontrarXYPlanta()`.

La funció `EncontrarXYPlanta()` fa un recorregut per tots els obstacles de l'escenari i en calcula, per a cadascun dels insectes a mesura que aquests es creen, la distància fins a cada objecte. Es queda amb la distància i la posició de l'obstacle més proper. D'aquesta manera es troba la planta més propera a un insecte. Finalment queda determinar a quin punt pertanyent a la planta s'ubicarà l'insecte. Aleatòriament es genera un angle que servirà per indicar la posició del perímetre on es col·locarà. Aplicant conceptes bàsics de trigonometria i coneixent el centre, el radi de l'obstacle i un angle, es determina una posició de la vora de la planta, que serà la posició inicial de l'insecte.

En aquest cas, tot i que el codi sigui menor que el de la funció `GenerarXY`, ja queda suficientment explicat amb la definició anterior, pel

que es veurà com s'utilitzen els dos mètodes per a obtenir-ne els punts on posicionar-hi els insectes. S'observa que s'utilitza tant la funció GenerarXY, com EncontrarXYPlanta, mentre que a la distribució dels robots tan sols s'utilitzava la primera.

```

/*****
/* Crear i ubicar els punts vermells
*****/
for(i=0; i<dlgEscenario.m_pointred; i++) {
    GenerarXY(m_distribucio_punts,&NormalGenerator1X,&NormalGenerator1Y,
    &NormalGenerator2X,&NormalGenerator2Y, &rand_x,&rand_y);
    EncontrarXYPlanta(&rand_x, &rand_y);
    p = CPoint( rand_x, rand_y);
    CRobotForage::AddItem(p);
    CRobotForage::AddColor(0);
}

/*****
/* Crear i ubicar els punts blaus
*****/
for(i=0; i<dlgEscenario.m_pointblue; i++) {
    GenerarXY(m_distribucio_punts,&NormalGenerator1X,&NormalGenerator1Y,
    &NormalGenerator2X,&NormalGenerator2Y, &rand_x,&rand_y);
    EncontrarXYPlanta(&rand_x, &rand_y);
    p = CPoint( rand_x, rand_y);
    CRobotForage::AddItem(p);
    CRobotForage::AddColor(1);
}

/*****/

```

Figura 9: Fragment de codi que descriu la col·locació inicial dels insectes.

4.2 Adaptacions a les estratègies de cerca

Tal com passava amb les ubicacions inicials de punts i robots, les estratègies de cerca també són modificables mitjançant l'entorn gràfic, en aquest cas, les opcions es mostren a la pantalla següent. Noti's que al costat de l'estratègia de cerca per una normal centrada en els N darrers punts hi ha un paràmetre que haurà d'introduir l'usuari. En cas de deixar-lo pel seu valor per defecte, es produirà una cerca aleatòria.

Tant les estratègies de cerca dels robots com les polítiques de reconfiguració es programen a la classe CrobotForage, que precisament és la mateixa classe dels robots del R&B Sim. En realitat, de *Foraging*, que consisteix en anar a buscar punts i portar-los a un indret, no se'n fa i s'ha mantingut el nom per coherència amb el R&B Sim, on tampoc es feia cap mena de captura dels punts. Al mètode Forage() s'hi implementa la màquina d'estats i s'hi poden trobar els estats de *Wander*, *Goto*, *Decide*, *Reconfig*, *Working* i *Pause*.

L'adaptació de les estratègies de cerca requereix modificar l'estat de *Wander*, en el qual el robot cerca els punts a trobar. Una vegada aquests punts ja estan al seu radi de detecció, passa a l'estat *Goto*, que provoca que el robot avanci en la direcció adequada segons el punt que s'ha detectat.

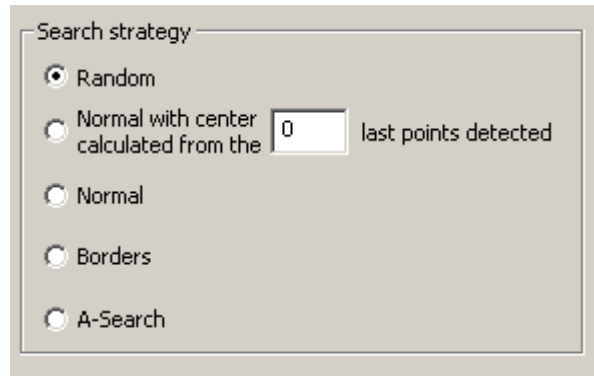


Figura 10: Opcions del simulador per a triar una estratègia de cerca.

Durant l'estat de *Wander*, entre d'altres aspectes rellevants, es crida a la funció `GetNextPoint()`, que segons l'estratègia de cerca proporcionarà un punt o un altre. Això es duu a terme mitjançant una crida a una altra funció anomenada `GenerarXY()` que actua de la mateixa manera que la funció que ubica els punts i els robots inicialment, excepte que aquesta incorpora l'estratègia *A-Search*. Només en el cas d'aquesta estratègia, la funció que proporciona el punt següent és `A-SearchGetPoint()`. En la resta dels casos ja s'ha vist com es generen els punts on desplaçar-se.

Pel que fa a `A-SearchGetPoint()` és una funció amb una complexitat molt més elevada que no pas la resta d'estratègies de cerca. El codi és força llarg i seria complicat incloure'l, però l'essència rau en calcular les distàncies de cadascuna de les plantes fins el robot. Una vegada finalitzat, s'ordenen les distàncies i el robot es desplaça al punt més proper que compleixi una sèrie de condicions. Aquestes condicions s'han posat per tal de fer que el robot mantingui un ordre a l'hora de visitar les plantes. Així, quan un robot es desplaça sempre anirà a visitar la planta que quedi a la seva dreta i que sigui més propera a la seva posició mentre continua avançant. Per tal d'aconseguir-ho el robot disposa d'informació (modelada com un seguit de variables) que indiquen la direcció del robot. Segons aquesta direcció, les condicions són unes o altres. Per posar tan sols un exemple, si el robot es desplaça cap a la dreta, es cercarà un punt amb una coordenada X major que la coordenada X de la posició del robot i amb una coordenada Y també major que la coordenada Y del robot.

Seguint aquest patró ordenat els robots col·lisionen menys i s'hauria de produir un estalvi energètic a més d'una cerca més efectiva dels punts.

Malgrat totes les avantatges que sembla tenir aquesta cerca enfront de les altres a l'hora de simular el comportament dels robots el funcionament de l'aplicació es veu molt afectat pel que fa al rendiment. Aquesta mena de cerca requereix ordenar moltes distàncies. En el cas d'un camp de 4 hectàrees, força petit en realitat, segons els patrons definits s'hi troben més de 300 plantes. Si a més, hi afegim 50 robots, per cadascun dels robots s'ha de calcular a cada pas de la simulació la distància fins a cadascuna de les 300 plantes. Això suposa una quantitat de càlcul important, tot i que s'hagi implementat un algorisme de cerca basat en el *quicksort*, més eficient que molts d'altres mètodes i, a més, que les distàncies no variaran massa d'un moment a un altre, doncs el moviment del robot és continu. Malgrat totes aquestes consideracions i optimitzacions, la simulació és molt més pesada que no pas el fet de generar un nombre aleatori que compleixi un seguit de propietats.

Malgrat haver obtingut un punt que s'hagi d'anar a trobar, no sempre serà possible arribar-hi. Això vol dir que un robot podria bloquejar el pas d'un altre que es dirigeix cap a un punt o fins i tot una planta ja visitada, podria interposar-se en el seu camí. Per evitar situacions com aquestes s'ha incorporat un mecanisme de detecció de robots atrapats.

Un robot a mesura que es desplaça va desant els valors de les coordenades de les seves darreres 50 posicions a cada pas del simulador. Una vegada s'ha acomplert aquest nombre d'iteracions, es comprova la distància entre la mitjana de tots aquests punts amb la posició actual del robot. Si aquesta distància és inferior a un valor, se suposa que el robot s'ha mogut menys del que seria raonable i, per tant, aquest robot està encallat. Per tal de desencallar-lo es tria un altre punt a buscar i es torna a l'estat de *Wander*, ja que aquest tipus d'encallament es pot produir tant a l'estat de *Wander* com a l'estat de *Goto*, quan un robot ja ha localitzat el punt a tractar.

Aquest sistema de detecció de robots encallats no és suficient quan es fixa una política de cerca A-Search. En aquest cas, com que el robot si busqués un altre punt, tornaria a trobar el mateix que l'ha deixat encallat, caldrà fer un canvi de direcció. Així, variant la direcció del robot en el sentit de les agulles del rellotge, cada vegada que es queda encallat i buscant posteriorment un altre punt, s'aconsegueix minimitzar la quantitat de robots encallats.

4.3 Adaptacions a les polítiques de reconfiguració

De la mateixa manera que en els apartats anteriors, les polítiques de reconfiguració també són modificables per l'usuari a l'entorn gràfic quan s'entra dins l'apartat de paràmetres de l'escenari. La ubicació d'aquestes opcions a paràmetres de l'escenari enlloc de paràmetres del robot s'ha fet per tal de respectar l'aparença i el comportament de l'aplicació R&B Sim. Les opcions, que ja s'han detallat en capítols anteriors són les següents:

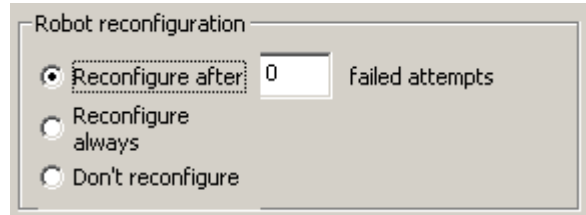


Figura 11: Opcions del simulador per a triar una política de reconfiguració.

Noti's, en aquest cas, que hi ha un paràmetre numèric per a l'estratègia de reconfigurar-se després d'un nombre determinat d'intents fallits a l'hora de trobar un insecte del mateix color. Si es deixa aquest paràmetre al seu valor per defecte, que és 0, el comportament serà idèntic a reconfigurar-se sempre.

Per adaptar aquest aspecte s'ha hagut de modificar l'estat de *Decide* de la funció *Forage()*. El R&B Sim utilitzava funcions diferents segons si es permetia la reconfiguració (funció *Forage()*) o no (funció *Track()*). En el nostre cas no farem distinció i s'ha eliminat del tot la funció *Track*.

Si hi ha reconfiguració sempre, l'estat *Decide*, procedirà a reconfigurar-se si el robot detecta un punt de color diferent o bé treballarà sobre el punt, si aquest és del mateix color que el robot.

En el cas de no haver-hi mai reconfiguració i trobar-se el robot un punt de color diferent, simplement passarà una altra vegada a l'estat de *Wander*, mentre que si no, hi treballarà passant a l'estat *Working*.

Si hi hagués l'opció de reconfigurar-se després d'un nombre d'intents fallits en detectar els punts, s'anirà acumulant cada intent en una variable del robot de manera que, en arribar a la cota marcada es reconfigurarà. Si es trobés abans amb un punt del seu mateix color, aquest comptador tornaria a 0, tornant-se a iniciar la seqüència.

En qualsevol dels casos anteriors es comptabilitza el temps necessari per a detectar el punt i, en el cas de reconfigurar-se, de tornar a detectar la posició del punt d'una altra mena. Això equival a una primera part del treball i resulta força costós tant en temps com en energia. Sembla, a priori, que una política de reconfiguració nul·la equivaldrà a menys punts tractats per una despesa energètica equivalent.

4.4 Diagrama de la màquina d'estats adaptada

Als apartats anteriors ja s'han anat introduint els termes de màquina d'estats i s'ha descrit alguna adaptació que s'hi ha fet per tal de modelar el comportament dels robots dedicats a eliminar insectes. En aquest sentit els canvis més importants es donen als estats de *Wander*, *Goto* i *Decide*.

En el cas de *Wander* i *Goto*, els estats encarregats de determinar el moviment del robot l'afegit més important consisteix en una detecció de col·lisions, que farà tornar el robot a l'estat de *Wander* en cas de detectar-se.

Pel que fa a la funció *Decide*, el robot durà a terme una primera inspecció de la planta cercant els punts que sap tractar. Malauradament aquesta inspecció també comporta una primera part del treball. Si el robot tracta erugues, haurà de sacsejar la planta i cercar les erugues al terra; si el que fa és detectar corcs, analitzarà les beines. Això és equivalent a dir que el robot fa una primera part de la tasca sense saber si el punt és del seu color o no, pel que conceptualment es pot dividir l'estat de *Working* en dues parts segons si el que es fa és identificar i localitzar els punts o bé eliminar-los.

Finalment l'R&B Sim contemplava l'opció de simular microrobots que, per tal de reconfigurar-se, necessitaven la presència de robots de color diferent a un radi inferior al radi d'abast del robot. En el cas de robots com els que s'han triat, fer això seria una pèrdua de temps ja que tenen la capacitat per tenir emmagatzemats els programes necessaris per a l'eliminació dels insectes de qualsevol dels dos tipus. És per això que l'estat que s'anomenava *Pause*, en el que els robots emetien la seva configuració, en el nostre cas ha desaparegut.

Un esquema simplificat de la nova màquina d'estats que reflecteix tots els canvis esmentats es detalla a continuació. Comparant-lo amb el diagrama d'estats original del R&B Sim, en manté l'estructura al màxim possible.

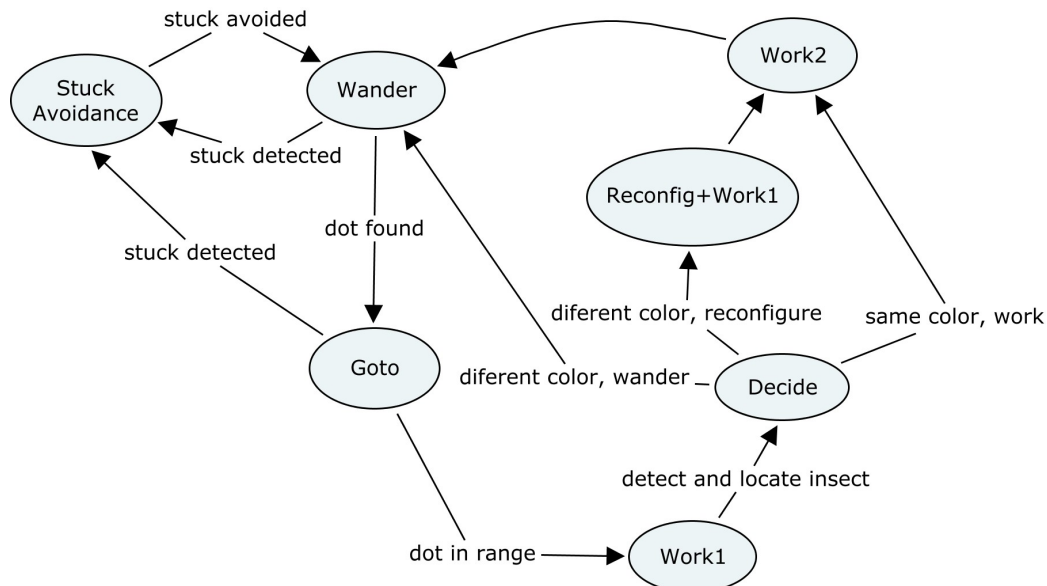


Figura 12: Graf d'estats dels robots eliminadors d'insectes.

El robot inicialment es troba a la recerca de punts a l'estat *Wander*, en cas de detectar-ne, passa a l'estat *Goto*, i quan té el punt a l'abast per treballar-hi, hi fa la primera part del treball, que consisteix en detectar si hi ha el tipus d'insecte que mata i localitzar-lo. En aquest cas, si el punt és del seu mateix color, es passa a la segona part del treball passant per l'estat *Decide*, però si no ho és cal identificar si cal reconfigurar-se. En cas de ser necessari fer-ho (segons la política de reconfiguració) es passa a l'estat de *Reconfig* que inclou tornar a desenvolupar la primera part de la feina per als altres tipus d'insectes. Finalment es fa la segona part de la feina. Si no es permetés la reconfiguració, es tornaria a l'estat de *Wander*, igual que en el moment en el que el robot es desplaça i no pot seguir avançant, tot i que en aquest cas passa abans per un estat d'e sorteig d'obstacles que farà canviar el següent punt a cercar.

Aquesta representació no vol dir que s'hagin programat aquests estats com a entitats independents, sinó que el robot es comporta d'aquesta manera combinant els estats que s'han explicat en aquest capítol.

Capítol 5

Simulacions

Per poder determinar l'eficàcia de cada comportament social és necessari realitzar una sèrie de simulacions de tal manera que es pugui comprovar el funcionament dels robots sota determinades circumstàncies. En aquest cas, les circumstàncies ja han estat descrites i es tracta d'un escenari realista corresponent a un camp de cigroneres de 4 hectàrees infestat per 200 insectes, un centenar de corcs i un centenar d'erugues. En aquest mapa s'hi ubiquen 25 robots multifuncionals capaços de matar corcs des del començament i 25 robots de la mateixa mena capaços de matar erugues. Tant els insectes com els robots s'han col·locat a les vores del camp, d'aquesta manera es simula un escenari corresponent a una invasió en les seves primeres fases, però amb una quantitat d'insectes molt elevada.

Les opcions de reconfiguració que admet el programa són principalment 3. L'una consisteix en permetre la reconfiguració dels robots en qualsevol moment, una altra no la permet en cap cas i una tercera depèn del nombre d'insectes de color diferent a l'esperat.

En canvi, d'estratègies de cerca en tenim 5 de diferents. Una que permet fer cerques de punts aleatòriament. N'hi ha també unes altres tres que fan que els robots es desplacin a punts que compleixen certes característiques. En concret, disposem de cerca de punts segons una normal de centre el mateix centre del camp, però desplaçat cap al sud. També existeix la cerca a les vores, on habitualment s'hi troben els insectes. Però també tenim dos tipus de cerca força diferents. Un que depèn dels darrers punts trobats i, en haver-ne trobat un nombre determinat, busca punts al voltant d'aquests seguint una normal. I un altre mètode que utilitza una sèrie de normes de circulació per tal d'evitar xocs i col·lisions entre insectes i garantir una millor cobertura del terreny, anomenada A-Search.

En total, s'han realitzat 15 simulacions seguint diferents paràmetres. D'aquesta manera definim 15 estratègies diferents i en podem analitzar els resultats des de diferents vessants, des de l'efectivitat eliminant insectes fins a la quantitat d'energia que han necessitat per aconseguir la seva fita. Tot seguit es detalla quina és cadascuna de les simulacions que s'han dut a terme.

Estratègia	Política de cerca	Política de reconfiguració
Str01_SR_RA	Aleatòria	Reconfigurar sempre
Str02_S3N_RA	Normal amb centre als 3 darrers valors	
Str03_SN_RA	Normal	
Str04_SB_RA	Cerca a les vores	
Str05_SA_RA	A-Search	
Str06_SR_RN	Aleatòria	No reconfigurar mai
Str07_S3N_RN	Normal amb centre als 3 darrers valors	
Str08_SN_RN	Normal	
Str09_SB_RN	Cerca a les vores	
Str010_SA_RN	A-Search	
Str011_SR_R3F	Aleatòria	Reconfigurar després de 3 intents fallits
Str012_S3N_R3F	Normal amb centre als 3 darrers valors	
Str013_SN_R3F	Normal	
Str014_SB_R3F	Cerca a les vores	
Str015_SA_R3F	A-Search	

Com pot observar-se, s'han dividit les estratègies en 3 blocs de 5. Cadascun d'aquests blocs es caracteritza per la política de reconfiguració triada. D'aquesta manera s'observarà el comportament diferent de cada política de cerca segons la política de reconfiguració, que en alguns casos és molt acusat.

5.1 Estudi de l'efectivitat: nombre de punts treballats

Observant el nombre de punts sobre els quals han treballat tots els robots de cada estratègia s'han obtingut les dades que es recullen al gràfic corresponent a la Figura 13.

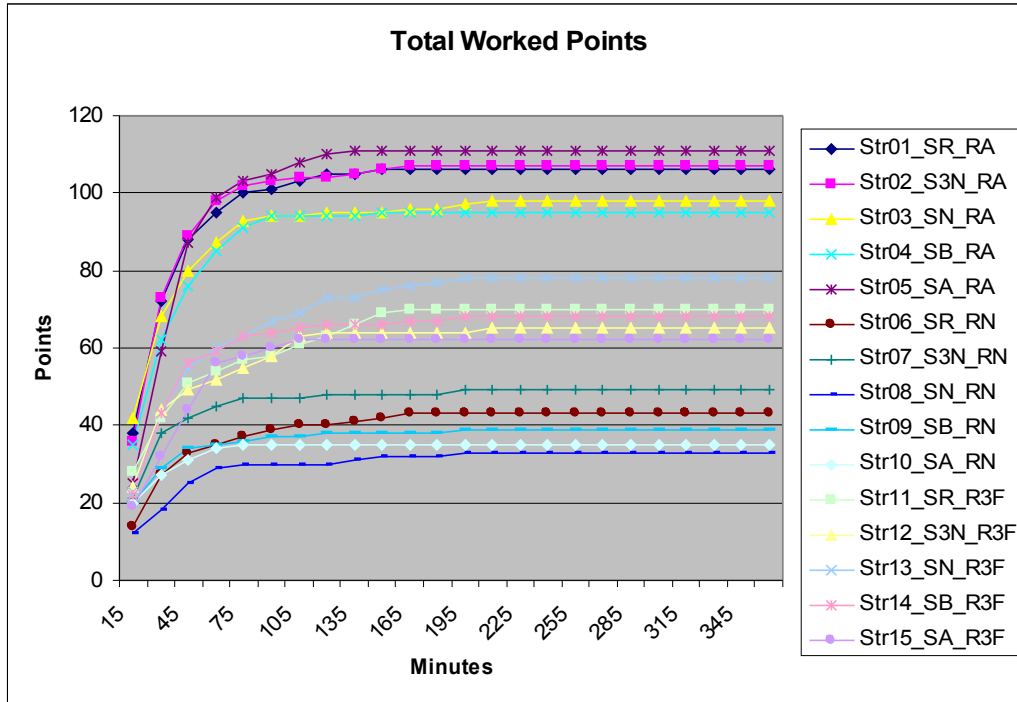


Figura 13: Nombre de punts eliminats en cada etapa de la simulació.

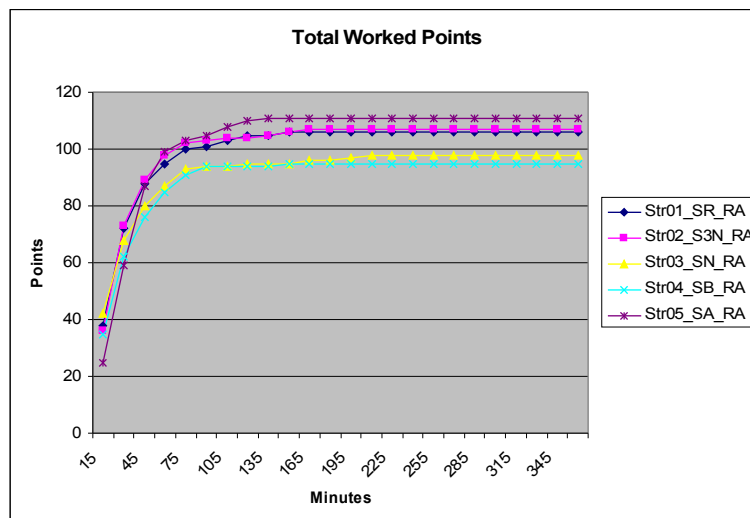


Figura 14: Nombre de punts eliminats per a les estratègies 1-5.

Per tal de comparar amb facilitat l'efectivitat de cadascuna de les estratègies, s'han desglossat els gràfics separant les estratègies segons la política de reconfiguració. D'aquesta manera es pot obtenir el gràfic de la Figura 14.

Del gràfic anterior, se'n pot deduir que l'estratègia que més insectes aconsegueix eliminar és l'estratègia Str05_SA_RA, basada en l'A-Search, que ha aconseguit eliminar un total de 111 insectes, el que equival a un 55,5% del total. De qualsevol manera, les diferències amb l'estratègia Str01_SR_RA i Str02_S3N_RA són força ajustades. A més, s'observa que la majoria dels punts es troben durant els primers 150 minuts, pel que es podria estalviar energia si s'acabés el treball en aquest instant. Si considerem les dades en global, aquesta és la política de reconfiguració que aconsegueix millor resultat amb molta diferència. Però abans de decantar-nos per aquest grup cal estudiar en detall els consums.

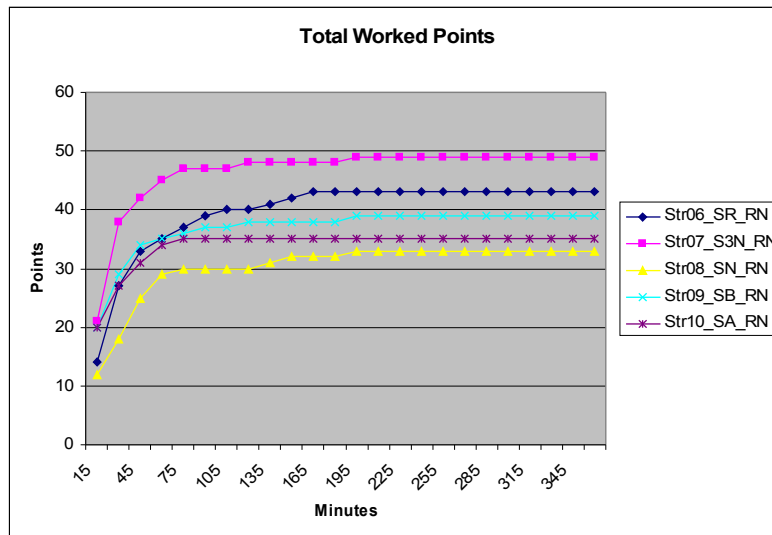


Figura 15: Nombre de punts eliminats per a les estratègies 6-10.

La figura anterior mostra les estratègies amb una política de no reconfiguració. El nombre de punts sobre els que es treballa és molt més reduït que en el cas anterior. Això segurament serà degut a la quantitat d'energia que es gasta i en la dificultat de trobar els punts, però aquesta teoria es comprovarà a l'apartat següent. És per això que el nombre més alt d'insectes eliminats és menys de la meitat que els valors obtinguts amb una política de reconfigurar-se sempre. En aquest cas, l'estratègia S10_SA_RN, l'A-Search, obté uns resultats molt més dolents que al cas anterior, mentre que l'estratègia basada en la cerca al voltant dels 3 darrers punts, anomenada Str07_S3N_RN trobats és la que ofereix un millor resultat.

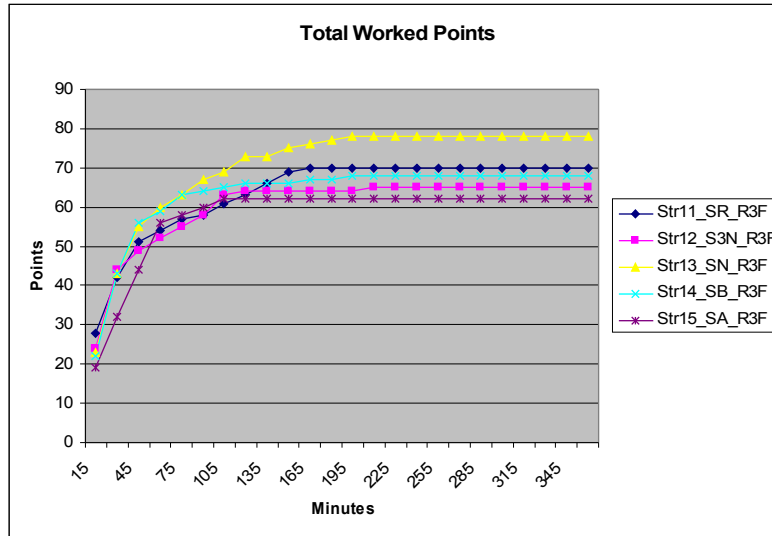


Figura 16: Nombre de punts eliminats per a les estratègies 11-15.

Quan s'analitzen les estratègies basades en una política de reconfiguració consistent en reconfigurar-se quan s'han trobat tres punts de color diferent al del robot els pitjors resultats els obtenim amb l'estratègia Str15_SA_R3F, basada en l'A-Search. En canvi, l'estratègia Str13_SN_R3F, amb una cerca normal, que s'havia comportat de forma molt deficient amb les anteriors polítiques de reconfiguració, s'ha vist molt beneficiada.

Resulta sorprenent que la política de cerca aleatòria ha obtingut bons resultats en els tres casos analitzats. De moment, si s'hagués de triar una estratègia per assolir la màxima efectivitat eliminant insectes, triaríem la cerca A-Search reconfigurant-se en cada ocasió que sigui necessari, és a dir, la Str05_SA_RA.

També és digne de destacar que l'estratègia basada en cercar els insectes a les vores no ha destacat especialment en cap de les 3 proves realitzades, contràriament al que es podria pensar. Això podria ser degut a la quantitat de xocs amb plantes o amb d'altres robots que dificulten que el robot acabi arribant a les vores, per això els punts, que principalment es troben als extrems, no s'acaben trobant.

5.2 Consum: anàlisi de la despesa energètica mitjana acumulada dels robots

Una vegada ja s'ha vist l'efectivitat de cada estratègia és necessari veure la quantitat d'energia necessària per eliminar els insectes que requereix cadascuna de les estratègies. La Figura 17 representa el consum acumulat promig dels robots de cadascuna de les estratègies.

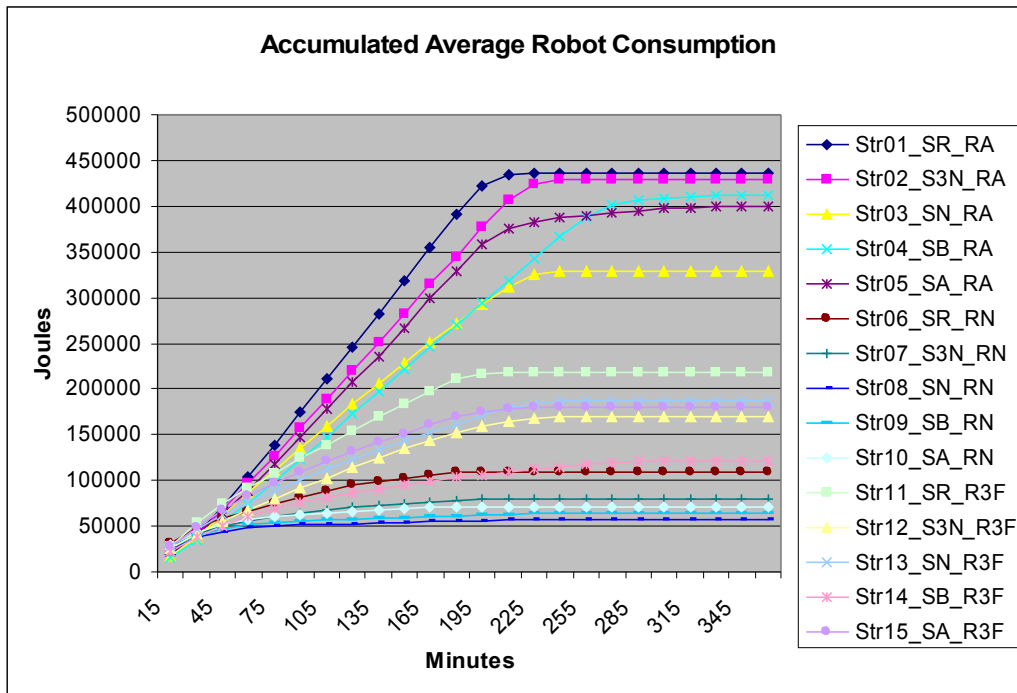


Figura 17: Despesa energètica acumulada de cadascuna de les estratègies.

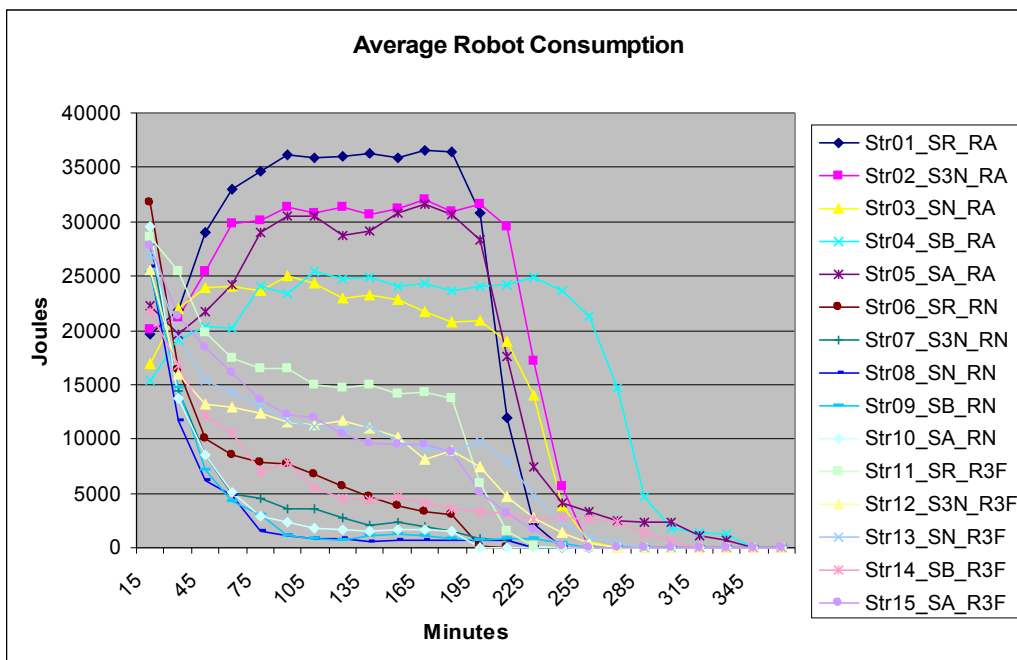


Figura 18: Despesa energètica de les estratègies.

La simulació de les estratègies s'ha aturat quan cap dels robots efectuava cap acció en un període raonable de temps. Hom podria pensar que això succeeix quan l'energia s'ha esgotat, però tots els robots tenen la mateixa quantitat d'energia inicial a les bateries per qualsevol de les estratègies. Així doncs, com és possible que el consum energètic surti diferent? La resposta l'hem de buscar en el fet que, quan un robot es queda sense energia roman aturat bloquejant el pas d'altres robots. Així, al final alguns robots, els que esgoten l'energia més ràpidament, es queden aturats aviat i bloquegen a la resta deixant-los tancats a la rasa a la que es troben. Poc a poc acaben esgotant la seva energia i deixant bloquejats d'altres robots. Al final, hi ha robots que no es desplacen perquè no poden moure's en cap direcció i no pas perquè no tinguin més energia.

Fixant-nos en la inclinació de les gràfiques anteriors és possible determinar el consum energètic. Tot i que abans s'havia vist que les estratègies basades en cerca aleatòria es comportaven força bé, també són les que tenen una despesa més elevada en cadascuna de les seves classes.

A més, en les polítiques sense reconfiguració el consum energètic és molt baix. Però per tal de comparar-les de forma adient serà necessari desglossar el gràfic tal com hem fet en els anteriors per tal d'apreciar-ne les subtilitats.

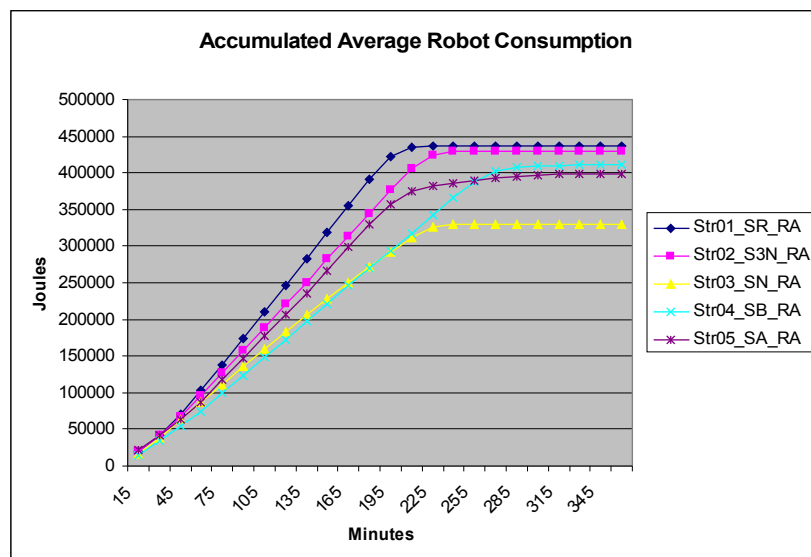


Figura 19: Despesa energètica acumulada a les estratègies 1-5.

En aquest primer gràfic, on queden plasmats els consums mitjans de les estratègies basades en una política de reconfiguració consistent en reconfigurar-se sempre, s'aprecia que el consum més elevat es dona amb l'estratègia Str01_SR_RA. Aquesta estratègia és la que consisteix en anar cercant punts de manera aleatòria. La majoria de les altres estratègies, però, tenen un comportament similar pel que fa a consum energètic. En canvi, l'estratègia Str03_SN_RA en surt força beneficiada, assolint un consum d'uns 325.000 Joules enfront dels 440.0000 de l'estratègia 1.

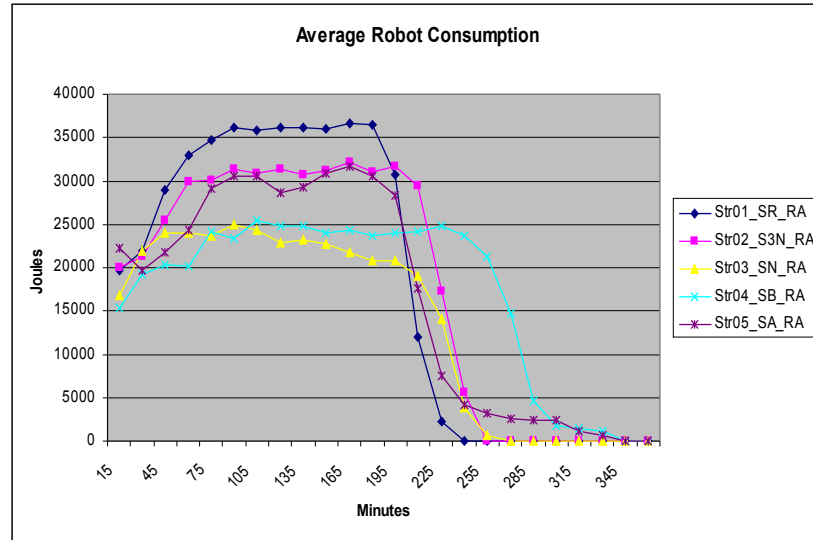


Figura 20: Despesa energètica a les estratègies 1-5.

L'estructura que segueix el gràfic anterior suggereix que pràcticament totes les estratègies segueixen el mateix patró pel que fa a consum d'energia i el seu consum més elevat es fa més o menys al mateix temps. L'estratègia Str04_SB_RA és la que sembla tenir més activitat durant més temps. Si ens adonem de que la majoria dels punts s'han capturat abans d'arribar als 135 minuts aproximadament, com pot comprovar-se a la Figura 14, es podria estalviar molta energia si s'aturés l'eliminació d'insectes en aquest punt.

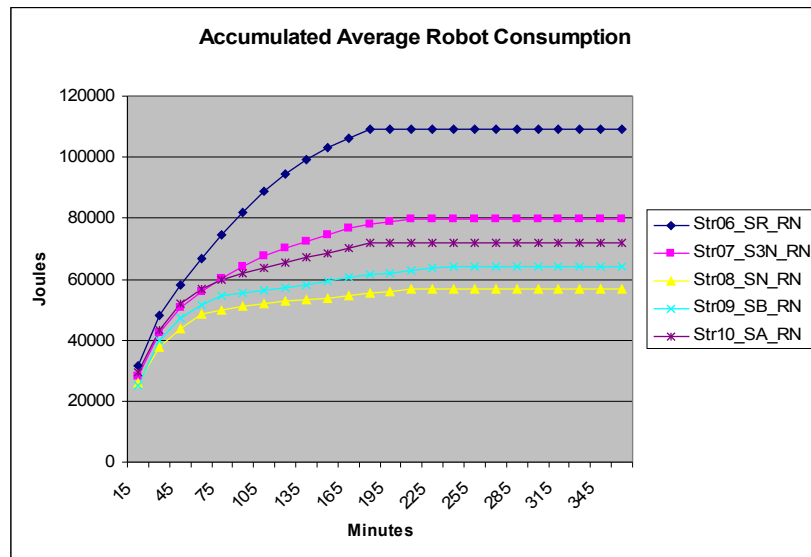


Figura 21: Despesa energètica acumulada a les estratègies 6-10.

Si prenem les dades de les estratègies quan no existeix una reconfiguració, l'energia consumida cau de forma significativa respecte les estratègies descrites anteriorment. Com que no hi ha reconfiguració, la despesa energètica és molt menor i això pot comprovar-se fins i tot en el pitjor dels casos, amb l'estratègia Str01_SA_RA, que obté 110.000 Joules per robot d'energia consumida. En canvi, polítiques que en punts eliminats donen millor resultat, com l'estratègia Str07_S3N_RN, aquí es comporten millor. En aquest cas, si no es permetés la reconfiguració, l'estratègia Str07_S3N_RN, basada en una cerca normal amb un centre format per la mitjana dels tres darrers punts, seria una bona alternativa ja que permetria eliminar més insectes consumint menys punts.

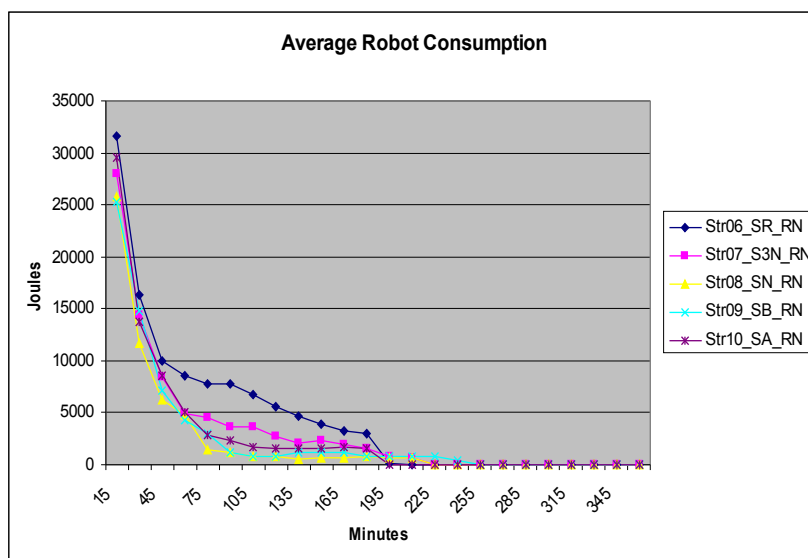


Figura 22: Despesa energètica a les estratègies 6-10.

En aquesta gràfica, que complementa la figura anterior, s'observen els mateixos resultats. Noti's que l'estratègia Str06_SA_RN té el consum més elevat durant tota la simulació, mentre que la Str07_S3N_RN és la que té un comportament més òptim. Aquesta conclusió és la mateixa que s'ha obtingut a la gràfica anterior.

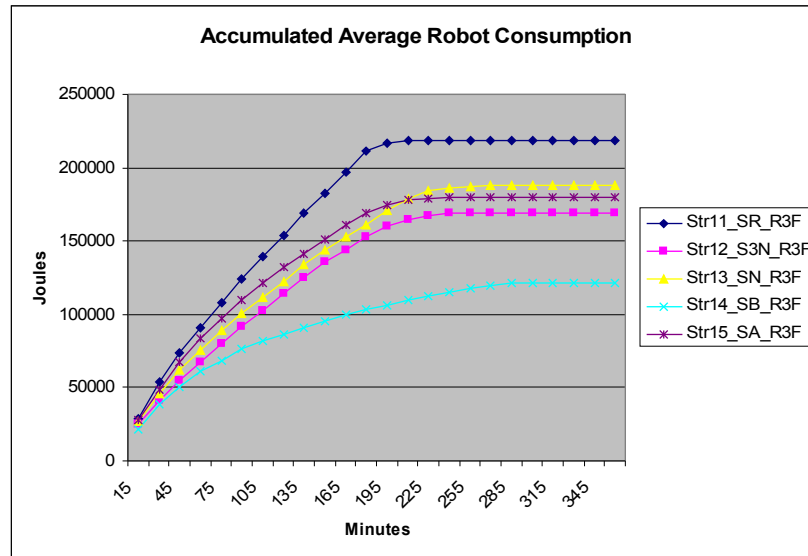


Figura 23: Despesa energètica acumulada a les estratègies 11-15.

Pel que fa a les estratègies amb reconfiguració després de 3 intents fallits, trobem que el màxim consum el tenim amb l'estratègia aleatòria, la Str11_SA_R3F. En aquest cas la diferència no és molt acusada, excepte en el cas de l'estratègia Str14_SB_R3F, que a més, com s'observa a la Figura 16 aconsegueix eliminar un nombre força elevat d'insectes amb un consum molt inferior al de la resta. Per tant, en aquest cas buscar a les vores és una bona opció. La resta d'alternatives tenen un consum similar, i només seria digne d'esment que l'estratègia Str13_SN_R3N aconsegueix matar més insectes que la resta, tot i que es manté un cost energètic elevat.

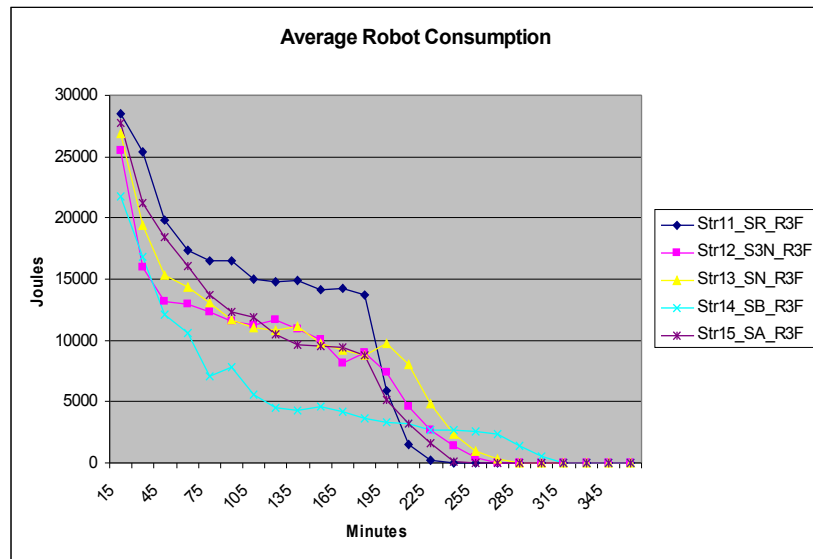


Figura 24: Despesa energètica a les estratègies 11-15.

En aquest cas veiem el pic que es produeix amb l'estratègia 11 d'una forma molt acusada. D'aquesta manera, el consum queda força elevat. La pujada que s'observa als 200 minuts per a l'estratègia 13 es deguda a que es manté eliminant insectes durant més temps que la resta.

5.3 Nombre de reconfiguracions: causa de les diferències de consum energètic

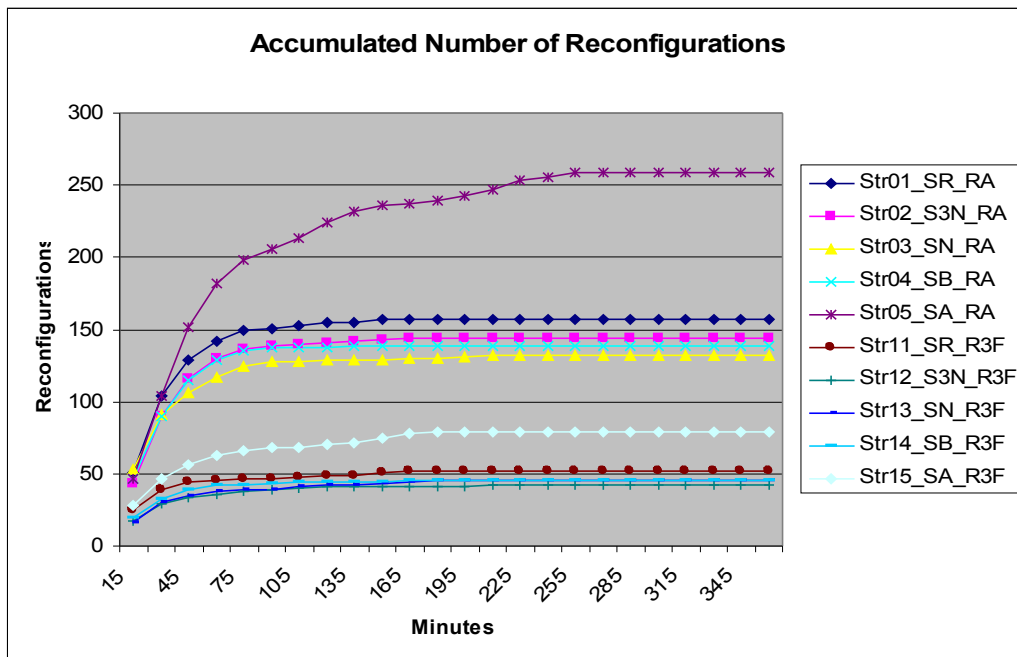


Figura 25: Nombre acumulat de reconfiguracions.

En aquest cas, no cal detallar més el gràfic ja que tots els valors són força similars i no és necessari tenir més precisió. Això és degut a que les estratègies que no presenten cap reconfiguració s'han obviat, ja que els seus resultats serien sempre 0.

De qualsevol manera en veure aquest gràfic el que sobta és la quantitat de reconfiguracions que duu a terme Str05_SA_RA, que supera amb escreix les reconfiguracions de l'estratègia Str02_S3N_RA o l'estratègia Str01_SR_RA. No obstant això, a la Figura 19 comprovem que l'estratègia Str02_S3N_RA i la Str01_SR_RA tenen una despesa energètica superior, cosa que resulta difícil d'explicar. Observant les dades obtingudes amb més detall s'entreveu la resposta a aquest enigma. Resulta que a l'estratègia 5 hi ha alguns robots que es reconfiguren un nombre inusual de vegades, però que maten a pocs insectes. Així, dediquen la seva energia en estar-se contínuament reconfigurant enlloc de dedicar-la a desplaçar-se o a exterminar insectes. És per això que en general el consum és inferior, perquè aquests robots són pocs però en comparació realitzen un nombre molt elevat de reconfiguracions i així distorsionen les gràfiques.

Capítol 6

Conclusions

Les conclusions d'aquest estudi es divideixen en diversos apartats, es començarà per un comentari del treball realitzat analitzant tot el projecte en el seu conjunt des de les primeres fases. De la mateixa manera es podrà observar el seguiment temporal de les tasques. Posteriorment es debateran els resultats obtinguts i es presentaran les conclusions a les que s'ha arribat. Finalment es farà una valoració del treball realitzat, destacant les dificultats trobades i les limitacions d'aquest estudi.

6.1 Assoliment dels objectius

6.1.1 Treball realitzat

En aquest projecte s'han plantejat un seguit d'objectius que val la pena recordar quan ja entrem al darrer capítol. El treball consistia en extreure informació sobre els comportaments socials entre eixams de robots reconfigurables. Cal recordar, que aquesta informació resulta especialment útil en el camp de la microrobòtica, però degut a l'escassa oferta i utilitat actual d'aquest tipus de robots, s'ha preferit analitzar el comportament de robots de característiques i dimensions superiors.

D'aquesta manera s'ha modelat un problema de control de plagues de corcs i d'erugues en camps de cigroneres. Aquest problema podia solucionar-se de forma molt similar a com es soluciona el problema de joguina anomenat Red & Blue, pel qual ja existia un simulador.

Fent un estudi rigorós del problema dels corcs i les erugues als camps de cigroneres, que ha estat complicat degut a l'aparent poca relació entre l'enginyeria informàtica i el món agrícola, s'ha estat capaç de modelar un camp de cigroneres i el comportament de corcs i erugues durant les fases d'invasió del camp.

Una vegada recopilada tota la informació necessària sobre les plantes, els camps de cigrons, els insectes i la seva eliminació s'ha procedit a modelar un robot capaç d'eliminar ambdues espècies. Primer de tot s'han estudiat diverses alternatives per eliminar-los amb un robot reconfigurable, cosa que ha portat a les solucions triades per a cadascuna de les espècies. Per tal de modelar el robot, s'han triat alguns

components i se n'ha analitzat el consum i el temps necessari per fer desaparèixer els insectes segons els algorismes triats per eliminar-los.

Una vegada modelat el robot i determinat el consum, ha estat necessari determinar unes característiques del camp que permetessin efectuar les simulacions en un entorn similar al real. És per això que s'ha triat un escenari amb els punts i els robots ubicats a les vores, d'unes dimensions de 4 hectàrees i amb 100 erugues i 100 corcs. Tot i ser una aproximació, serveix per tal de verificar el funcionament de cadascuna de les estratègies que seguiran els robots.

Quan ja s'ha obtingut un escenari realista on fer-hi les simulacions, s'han hagut d'implementar diverses estratègies que determinen els comportaments dels robots. En concret, segons les possibilitats que s'han ofert al nou simulador, s'ha optat per combinar tres polítiques de reconfiguració diferents amb cinc polítiques de cerca. D'aquesta manera s'han obtingut 15 comportaments diferents que anomenem estratègies.

Executant el simulador i simulant el comportament dels robots mentre es desaven les dades obtingudes, s'han pogut extreure les dades de consum d'energia i d'insectes eliminats per a cadascun dels robots cada cert temps.

Finalment, ajuntant tota la informació obtinguda en les simulacions i durant les primeres fases del projecte. S'han extret les conclusions que es presenten al següent apartat.

6.1.2 Seguiment de la planificació temporal

La planificació temporal feta a l'inici s'ha respectat en la mesura del possible. Com molt sovint succeeix sempre hi ha imprevistos i aquest treball no ha estat cap excepció. La tria del problema va ser un procés llarg que es va endarrerir més del previst. A més, la recerca d'informació sobre plantes i plagues tampoc ha estat gens fàcil. El fet de triar una planta o una altra, d'estudiar-ne les plantes i la viabilitat per matar les plagues amb un robot, així com de la manera de fer-ho, ha trigat força més del temps previst, provocant en alguns moments un decalatge d'un mes.

Tot i els imprevistos, dedicant més hores i reduint el nombre de simulacions portades a terme, s'han aconseguit assolir totes les fites assenyalades. Per tot això es pot considerar la planificació tot un èxit, si partim de la base de que generalment les planificacions no s'acaben complint al cent per cent.

6.2 Anàlisi dels resultats i consideracions importants

Fent un anàlisi dels resultats obtinguts, les polítiques amb una millor eliminació d'insectes són les que permeten una major reconfigurabilitat. Malgrat aquest benefici, també són les que consumeixen més energia per fer la mateixa tasca, sent les que menys energia consumeixen les estratègies que permeten menys reconfiguracions.

A més resulta curiós com segons la política de reconfiguració, les estratègies amb polítiques de cerca diferents obtenen uns resultats tan dispars. I cal destacar que la política de cerca aleatòria està sempre entre les 3 més efectives sigui quina sigui la política de reconfiguració. L'única pega destacable de l'estratègia aleatòria és l'alt consum energètic que té.

En general, si no hi ha una forta limitació energètica serà preferible utilitzar l'estratègia de cerca A-Search amb una política de reconfigurar-se sempre, que correspon a l'estratègia Str05_SA_RA. En cas contrari, es podria optar per una estratègia més equilibrada quant a efectivitat i consum i es podria optar per l'estratègia Str04_SB_RA, consistent en una cerca a les vores i reconfiguració sempre, que assoleix bons resultats tot i que triga més temps en fer-ho.

Un dels problemes que s'han trobat i que han afectat directament els resultats és la presència al mapa dels robots que es queden sense energia. Una opció seria detectar quan un robot s'està quedant sense bateria i obligar-lo a desplaçar-se cap a un punt determinat del mapa. Així, de passada, també se solucionaria el problema d'haver-los de recollir. A més observant les gràfiques s'observa que a partir dels 200 minuts en la majoria dels casos ja no es troben nous insectes, pel que es podria fer retornar els robots estalviant un important cost energètic.

Les simulacions fetes durant les fases de desenvolupament han demostrat que el nombre de robots i d'obstacles al mapa és decisiu per determinar el comportament dels robots. Concretament l'estratègia de cercar punts a les vores en un camp sense cap mena d'obstacles on un robot es pot moure lliurement, el robot acaba visitant majoritàriament les zones limítrofes. Si en canvi es pren un mapa amb obstacles, com poden ser plantes o d'altres robots, s'observa que els punts tendeixen a xocar abans d'arribar al seu objectiu i finalment romanen pel centre del camp, en comptes d'anar cap a les vores.

Els comportaments socials, doncs s'han de comprovar fent les simulacions de l'escenari al complet, ja que extrapolar-ne els resultats sense tenir en compte algun factor podria distorsionar totalment el resultat.

6.3 Valoració final

Al llarg d'aquest projecte s'ha pogut comprovar teòrica i experimentalment que trobar una solució basada en robots distribuïts sense cap mena de control centralitzat és una tasca complicada. De fet, la majoria de sistemes multirobot que es poden trobar al mercat, que per posar només un exemple s'utilitzen molt a la indústria farmacèutica, utilitzen un sistema de control centralitzat.

De ben segur que tots aquests sistemes han estat dissenyats tenint en compte totes les opcions possibles, i també és segur que han analitzat la possibilitat de fer-ho mitjançant un sistema totalment distribuït. Això vol dir que difícilment aquest treball podria haver arribat a una millor solució que la que podria obtenir-se amb un control exhaustiu dels moviments dels robots i un coneixement detallat de l'entorn pel que es mouen.

Ara bé, no cal perdre de vista que l'objectiu del projecte no era pas trobar la millor solució al problema dels insectes als camps de cigroneres, sinó que tal com s'indica al títol del projecte, es tracta d'un estudi de comportaments socials dels robots. En aquest sentit, tot i no haver obtingut una gran eficàcia eliminant insectes, aquest treball permet comparar estratègies en un problema concret.

També cal tenir en compte, que es tracta d'un problema fictici, que si bé existeix, hi ha d'altres mètodes més efectius per solucionar-lo. Malgrat tot, el temps del que es disposava era limitat i posar-se en contacte amb

la gent del camp hauria estat una tasca massa llarga, que hauria redundat en una menor qualitat del treball, sense aportar-hi uns beneficis considerables, doncs l'estudi (amb certes adaptacions) podria servir per a qualsevol altre problema.

Precisament aquest treball s'havia plantejat per la seva possible aplicació directa amb eixams de microrobots que no podien funcionar de cap altra manera que no fos mitjançant la reconfiguració i el treball distribuït per la seva poca capacitat de càlcul i de memòria. Per tant, els resultats que s'han obtingut seran vàlids també en aquests casos per a problemes similars amb aquest tipus de robots.

El treball s'ha limitat als comportaments socials que podrien ser similars al joc del Red & Blue, cosa que podria semblar que limita les possibilitats de l'estudi. Però en realitat, tot i que a primera vista pugui semblar-ho, ha permès estudiar més en profunditat d'altres aspectes, com per exemple la tria del robot, els efectes de les plagues, la seva distribució, com s'eliminen, i una llarga llista de detalls que haurien estat difícils d'aconseguir en 450 hores de projecte. Per això es va acotar el problema limitant la recerca en un dels aspectes i prenent com a punt de partida un camí del qual no ens hem apartat al llarg de tot aquest estudi.

Les dificultats que ha comportat el treball han estat diverses i algunes ja s'han anat explicant a mesura que apareixen els temes als capítols anteriors. D'altres s'han enunciat en aquest darrer capítol. En qualsevol cas ha estat especialment difícil trobar informació sobre plantes i sobre plagues, un tema que normalment escapa al domini d'un enginyer informàtic. Per altra banda podria ser que en la recerca del robot no s'hagi encertat amb el model més adequat per dur a terme aquesta tasca, però s'ha fet tot el possible i s'han analitzat diverses opcions amb tot el temps del que es disposava. A més, el simulador R&B Sim disposava de documentació, però hi mancaven explicacions, a més de tenir funcionalitats que no s'havien implementat del tot, cosa que causava una sensació estranya i feia que les primeres simulacions no funcionessin adequadament.

Malgrat tot, ha estat una experiència molt enriquidora que toca diversos camps que de vegades no són tractats per un estudiant i constitueix un fil que acaba unint un bon grapat de disciplines que conformen la informàtica.

Referències

- [1] ARNÓ J. et al., "Seguimiento de las Poblaciones de *Helicoverpa Armigera* (Hübner) (Lepidoptera: Noctuidae) en cultivo de tomate. Bol. San. Veg. Plagas. 1994.
<http://www.mapa.es/ministerio/pags/biblioteca/plagas/BSVP-20-01-251-260.pdf> (21/05/2009).
- [2] "Bean Weevil". WIKIPEDIA. 2009.
http://en.wikipedia.org/wiki/Bean_weevil (21/05/2009)
- [3] BRUEMMER D. , "Robot Swarm", Idaho National Laboratory. 2009.
<http://www.inl.gov/adaptiverobotics/robotswarm/> (25/02/2009)
- [4] CHANDRASEKARAN S.; HOUGEN D.F., "Swarm intelligence for cooperation of bio-nano robots using quorum sensing". *Bio Micro and Nanosystems Conference, 2006*. BMN apos; vol. 06. 2006. pàg.104.
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F4129377%2F4126236%2F04129442.pdf%3Farnumber%3D4129442&authDecision=-203> (27/03/2009)
- [5] DRZEWIECKIM, T.M., SHUMAN, D., "Acousto-fluidic Detection of Insect Larvae in Grain". Acta Hort. (ISHS). 2001.
http://www.actahort.org/books/562/562_27.htm (24/07/2009)
- [6] FREITAS R.A., "Nanomedicine FAQ". Foresight Institute. 1998.
<http://www.foresight.org/Nanomedicine/NanoMedFAQ.html> (04/03/2009)
- [7] GARCIA F. et al., "Insectos Endopterigotos". *Plagas Agrícolas II*. Universitat Politècnica de València. 1989.
- [8] GARCIA S., "Manejo del Cultivo de Garbanzo". INTA EEA Salta. 2002.
<http://www.inta.gov.ar/salta/info/documentos/legumbres/garbanzo.pdf> (21/05/2009)

- [9] "Helicoverpa Armigera". WIKIPEDIA. 2009.
http://en.wikipedia.org/wiki/Helicoverpa_armigera (21/05/2009).
- [10] INFOAGRO.COM., "El Cultivo del Garbanzo". *The Chickpea Growing*. Infoagro Systems S.L.
<http://www.infoagro.com/herbaceos/legumbres/garbanzo.htm>
 (21/05/2009)
- [11] INFOAGRO.COM., "El Cultivo del Tomate". *The Tomato Growing*. Infoagro Systems S.L. <http://www.infoagro.com/hortalizas/tomate.htm>
 (21/05/2009)
- [11] MALIK J. et al., "Contour and Texture Analysis for Image Segmentation", Computer Science Division, University of California at Berkeley. 1999.
- [12] PALIWAL J. et al., "Insect species and infestation level determination in stored wheat using near-infrared spectroscopy", *Canadian Biosystems Engineering, Le génie des biosystèmes au Canada* vol. 46. Canadian Society for Bioengineering. 2004.
<http://home.cc.umanitoba.ca/~paliwalj/PaperPDFs/CBE%20NIR%20insects.pdf> (24/07/2009)
- [13] PUCHE H. et al., "Estimating Dispersal Rate of the Silky Cane Weevil (Coleoptera: Curculionidae) ", *Journal of Applied Entomology* vol. 129 núm 6. Blackwell. 2005. pp. 293-299. <http://cat.inist.fr/?aModele=afficheN&cpsidt=16938892> (21/05/2009)
- [14] RIBAS-XIRGO, L. et al., "A Framework for Reconfigurable Robot Swarm". *IX Workshop en Agentes Físicos*. 2008.
- [15] SAHIN E., "Swarm Robotics". KOVAN Research Lab, Dept. of computer Engineering, Middle East Technical University. 2009.
<http://www.swarm-robotics.org/> (25/02/2009)
- [16] SOUTHGATE B.J., "Manual sobre Insectos que Atacan a las Semillas de Acacia", *Coleoptera*. Food and Agriculture Organization of the United Nations. 1983. <http://www.fao.org/DOCREP/006/Q2585S/Q2585S03.htm>
 (21/05/2009)

- [17] VURAL H., KARASU A., "Variability studies in chickpea (*Cicer arietinum* L.) varieties grown in Isparta, Turkey". *Revista UDO Agrícola* núm. 7 pàgs. 35-40. 2007. <http://www.bioline.org.br/pdf?cg07006> (07/09/2009)
- [18] WÖRN H. et al., "Micromechatronics and Microrobotics Group". Institute for Process Control and Robotics (IPR). 2003. <http://i60p4.ira.uka.de/tiki/tiki-index.php?page=Applications> (25/02/2009)
- [19] WOERN H., "I-Swarm", institute for Process, Control and Robotic. 2008. http://www.ipr.ira.uka.de/iswarm/MainPage/Project/P_Overview1.htm (25/02/2009)
- [20] YUS R., "Las Especies de Brúquidos (Gorgojos de las Leguminosas) de Interés Agrícola y Fitosanitario". Bol. Serv. Plagas. 1976. <http://www.mapa.es/ministerio/pags/biblioteca/plagas/BSVP-02-01-001-035.pdf> (21/05/2009)

Resum. Els eixams de robots distribuïts representen tot un món de possibilitats al camp de la microrobòtica, però existeixen pocs estudis que n'analitzin els comportaments socials i les interaccions entre robots autònoms distribuïts. Aquests comportaments han de permetre assolir de la manera més efectiva possible un bon resultat. Prenent com a base l'objectiu esmentat, aquest treball detalla diferents polítiques de cerca i de reconfiguració dels robots i estudia els seus comportaments per tal de determinar quins d'ells són més útils per solucionar un problema concret amb les plagues d'erugues i corcs als camps de cigroneres.

Resumen. Los enjambres de robots distribuidos representan un sinfín de posibilidades en el campo de la microrobótica, pero existen pocos estudios que analicen los comportamientos sociales y las interacciones entre robots autónomos distribuidos. Estos comportamientos deben permitir alcanzar de la forma más efectiva posible un buen resultado. Tomando como base el objetivo mencionado, este trabajo detalla diferentes políticas de búsqueda y de reconfiguración de los robots y estudia sus comportamientos para determinar cuales de ellas son más útiles para solucionar un problema concreto con las plagas de orugas y gorgojos en los campos de garbanzos.

Abstract. Distributed robots swarms represent lots of new options in robotics, but there are very few studies concerning social behaviors and interactions among distributed autonomous robots. These behaviors should allow a good result to be achieved as effectively as possible by the robot swarm. While searching the mentioned goal, this project details several search and reconfiguration policies and studies the robots behaviors. The simulation of these strategies, will allow us to judge which of them are the most useful to solve one particular problem with plagues of caterpillars and weevils at chickpea fields.